Modeling and Formal Analysis of High-Assurance Mixed-Reality Systems

Isaac Amundson, Junaid Babar, Heber Herencia-Zapana,
Simone Fulvio Rollini, Ben Brussee
Collins Aerospace

Peggy Wu, Timothy E. Wang
RTX Technology Research Center

Amanda K. Newendorp, Adam R. Kohl, Stephen J. Fieffer, Shayama S. Khan, Mohammadamin Sanaei,
Mieszko Muscala, Stephen B. Gilbert, Eliot Winer, Michael C. Dorneich, James Lathrop
Iowa State University

David Musliner, Robert P. Goldman,
Jeremy Gottlieb
Smart Information Flow Technologies

Parth Ganeriwala, Candice Chambers,
Siddhartha Bhattacharyya
Florida Institute of Technology

***Abstract*—Mixed-reality (MR) systems are seeing increased deployment in high-assurance aerospace applications, necessitating rigorous analyses of the complex interactions with their human operators and the surrounding environment. Traditional verification methods often neglect operator behavior or assume overly simplistic interactions, leaving MR systems vulnerable to attacks involving human cognition. This paper introduces the Modeling and Analysis Toolkit for Realizable Intrinsic Cognitive Security (MATRICS), aimed at formally assuring MR systems against cognitive adversarial threats. MATRICS integrates cognitive, environmental, and device modeling techniques to comprehensively address four categories of cognitive attacks: physiological, perceptual, attentional, and status-based. Our preliminary models and verification efforts validate the feasibility of our approach to provide essential cognitive security assurance, thereby enhancing operational effectiveness in adversarial contexts.**

I. Introduction

The past decade has witnessed the rapid advancement of immersive system technologies, including the use of mixed-reality (MR) systems[1] in high-assurance aerospace applications (e.g., the F-35 helmet-mounted display [2]). To keep pace, we must explore novel methods for verifying that these systems—and their users—are protected from adversarial exploitation. Traditionally, design-time analysis of avionics systems either leaves out the user or makes assumptions regarding user behavior (e.g., expecting users to always respond to events following pre-defined procedures). This approach may be inadequate for mixed-reality systems because the user, system, and environment are so entwined that they must be modeled and analyzed together. Failure to do so leaves the human-machine system vulnerable to a variety of failure modes and threats. More sophisticated analyses require accurate models of human cognitive behavior, new formal analysis methods, and tools that provide the rigorous assurance needed for the safe and secure deployment of MR systems.

The US Department of Defense has recognized the need for advanced analysis techniques for mixed-reality system designs through the DARPA Intrinsic Cognitive Security (ICS) program, which studies the feasibility of applying formal methods to verify that users of tactical mixed-reality systems will be protected from adversarial *cognitive* attacks. This emerging class of attack exploits the intimate connection between users and mixed-reality devices. Such attacks are generally unimpeded by traditional security safeguards (such as those implemented in [3]) because rather than exploiting vulnerabilities in the MR device hardware/software or supply chain, these attacks directly manipulate the human operator. For ICS, our team is developing the Modeling and Analysis Toolkit for Realizable Intrinsic Cognitive Security (MATRICS), which facilitates the development of *provably secure* mixed-reality systems.

In this paper, we provide an overview of MATRICS and present initial models and formal analysis methods that demonstrate our approach. MATRICS methods and tools support an extensive combination of cognitive, environment, and device modeling formalisms that encompass a broad area of the cognitive attack space. We illustrate our approach with models of MR human-machine designs that include aspects of cognitive processes and system functions. We developed the models in the context of an ICS-relevant mission and analyzed them to prove guarantees covering four distinct categories of cognitive attack:

1) Physiology -– in which an adversary applies stimuli to cause harm to the MR operator (e.g., nausea, headaches, etc.)
2) Perception -– in which an adversary causes the MR

[1]Although this work primarily focuses on augmented-reality (AR) systems, in which the real world is overlayed with digital information, we use the more general term *mixed-reality* following the taxonomy described in [1] because our methodology is not just limited to AR.

operator to misapprehend either real-world or virtual information (e.g., using bright light to wash out information on a digital display)

3) Attention -– in which an adversary distracts the MR operator (e.g., flooding an observation zone with decoys to increase visual search demand)
4) Status — in which an adversary gains unauthorized access to confidential operator data captured by the MR system (e.g., tracking blink rate to determine when the user is bored or fatigued)

For the design of MR aerospace applications, we envision MATRICS tools and methods supporting the workflow illustrated in Fig.1.

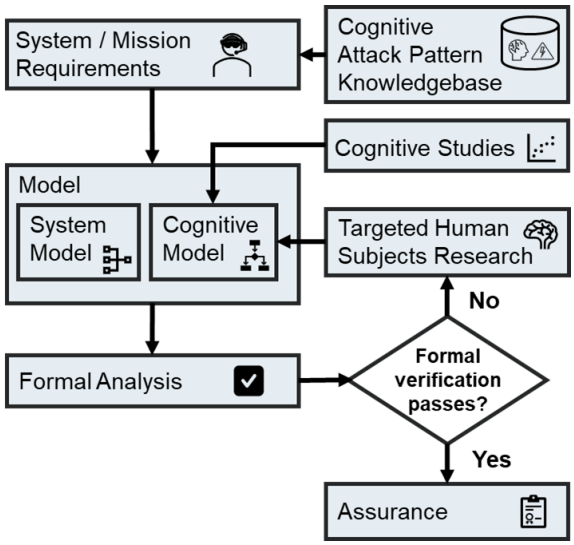

Fig. 1. Formal assurance workflow for mixed-reality systems.

Within the workflow, a preliminary cognitive cybersecurity assessment is conducted by iterating over a list of known vulnerabilities and attacks and producing new cyber requirements to address them when applicable. These, along with other system-level requirements, drive the refinement of a system model that includes aspects of the MR device hardware, software, and the mission environment. The model also includes relevant operator cognitive behavior from the cognitive science literature. Formal analysis is then performed on the model to provide assurance that the MR system design is protected from the cognitive attacks identified during the cyber assessment. It may be the case that the existing cognitive data used to inform the model is approximate (e.g., collected in virtual reality environments rather than augmented reality) or insufficient to prove cognitive guarantees. In this case, targeted human subjects research will be required to collect data that can produce a high-fidelity cognitive model conducive to formal reasoning.

Efforts to build system models that include human operator behavior and formally verify safety properties go back decades. Formal models of human operators have been developed using general languages such as process algebra, Petri-Nets, and more domain-specific ones such as task analytic and cognitive modeling languages. A detailed survey of models for the verification of human-machine systems can be found in [4] and a comprehensive handbook of formal methods for human-computer interaction can be found in [5]. Although MATRICS has some foundational similarities with these previous efforts, formal verification of cognitive attack protection in MR systems is a new domain requiring novel approaches, such as those described in the remainder of this paper. In Section II, we present our cognitive attack pattern knowledgebase for eliciting cognitive guarantees and corresponding design mitigations. We then introduce an example MR mission scenario in Section III and use it to illustrate modeling and formal analysis of multiple categories of cognitive attacks. We conclude in Section IV by listing planned research activities that will enable the eventual transition of MATRICS technologies into real-world aerospace product development workflows.

II. Repository of Cognitive Attack Patterns

Effective protection against cognitive attacks in mixed-reality systems requires an understanding of how an adversary may exploit cognitive vulnerabilities. Currently, there is no public resource that captures and classifies this information. We are therefore building a publicly-accessible Repository of Cognitive Attack Patterns (ReCAP), similar to MITRE's Common Attack Pattern Enumeration and Classification (CAPEC) [6], but specific to the cognitive security domain. Online publication[2] of the knowledgebase will provide a valuable tool to future mixed-reality aerospace system developers, independent of the formal rigor they use to develop and analyze their systems.

A central challenge for ReCAP is designing a platform capable of effectively categorizing, storing, and presenting cognitive attack data to support cognitive security analysis and community collaboration. To address this challenge, we have adopted a structured two-step approach. The first step involves identifying and defining the key entities relevant to MR system security, including attacks, vulnerabilities, and defense mechanisms. The second step focuses on mapping these entities to a relational database schema and designing the corresponding web interface and infrastructure to display and manage this information. The following subsections provide an explanation of these two steps.

A. Attacks, Vulnerabilities and Defenses

A high-level view of our cognitive attack taxonomy is illustrated in Fig. 2. A *cognitive attack* occurs when an attacker manipulates or disrupts a MR system and/or its human operator, resulting in interference with the system, its users, or the tasks they are performing. This manipulation is made possible by exploiting *vulnerabilities* within the MR system or its operators. Vulnerabilities are weaknesses in the MR components, user interactions with these components, or user behaviors that can be exploited by attackers to achieve their

[2]https://github.com/loonwerks/ReCAP

objective. The components of the MR system targeted in the attack due to these vulnerabilities are referred to as *attacked entities*. The *consequences* of such an attack may include a decline in the system's integrity, operator performance, or the overall user experience. A MR attack takes place within a specific context, known as the *environment*, which defines the setting in which the attack occurs. The core dynamic in this process involves the attacker executing a series of steps, known as an *attack sequence*, to exploit the vulnerabilities in the attacked entity. These steps may include manipulating the system's components or influencing the user's behavior.

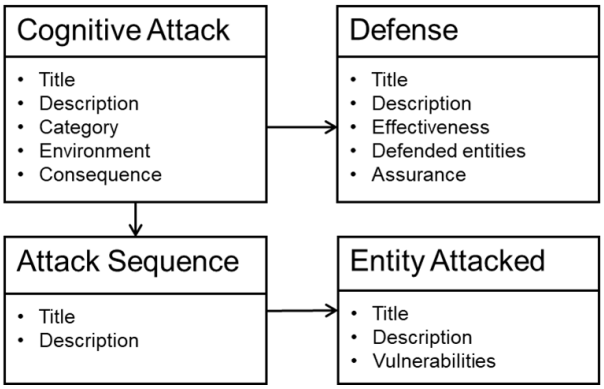

Fig. 2. ReCAP taxonomy.

To address attacks and vulnerabilities, especially those involving user cognition and interaction, cognitive *defense* mechanisms are introduced. Cognitive defense refers to targeted actions aimed at mitigating or eliminating the exploitation of MR system features that could be leveraged to carry out an attack. The *effectiveness* of these defenses is measured by their ability to reduce or neutralize the attacker's ability to exploit specific vulnerabilities. *Assurance* objectives, when satisfied, build confidence that a given defense mechanism will provide adequate protection against the cognitive attack.

B. Data Base and Web Page Design

The conceptual entities defined above were mapped into a structured relational database schema and we developed a corresponding web-based interface for storing, managing, and visualizing the cognitive attacks. A relational SQL database organizes the information using a set of interrelated tables, each representing a key entity involved in MR attacks, vulnerabilities, or defense mechanisms. With data organized in this structure, the web interface is presented dynamically through database-driven views. These views reveal key connections, such as links between specific vulnerabilities and attacks. Robust search and filtering tools allow users to explore the dataset by MR features, vulnerability types, or defense effectiveness.

Although we are still in the process of validating the cognitive attack classification schema and user interface, we believe ReCAP has been designed to provide the relevant insights needed for cognitive cyber-requirement elicitation. Contributions to ReCAP from the broader research community are encouraged in order to maximize utility of the knowledge-base and build consensus on its contents.

III. Formal Design and Analysis

In this section, we describe the modeling and analysis methods used by MATRICS. To illustrate our analytic approach, we first describe a scenario involving a helmet-mounted display (HMD) application to identify people carrying suspicious packages at border crossings, airports, and other security checkpoints. The system highlights potential malicious agents by drawing a red bounding box around them, and the HMD operator (e.g., a checkpoint guard in an observation tower) must then visually check to see whether the highlighted person is indeed a threat. If so, the guard will confirm the alert, which may then trigger ground personnel to intercept the package or cause the checkpoint gate to close. Otherwise, the guard will dismiss the alert, which will remove the bounding box. The guard can also alert to a suspicious person that was not identified by the scanning system. The scenario is illustrated in Fig. 3.

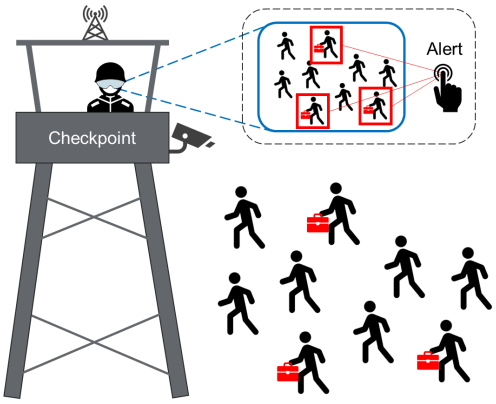

Fig. 3. Example mission scenario.

A cognitive security assessment, informed by the cognitive attack patterns in ReCAP, indicates that the following cognitive attacks could be possible:

- Physiology: The adversary overloads the HMD processor, increasing latency and decreasing framerate of the display until the guard feels nauseous and removes the headset. Once the HMD is removed, the adversary sends more contraband through.
- Perception: The adversary flashes an intense light at the guard, causing temporary pupil constriction that results in missed alerts.
- Attention: The adversary prepares a crowd of decoy agents, taxing the guard's visual search demand and causing the guard to be fixated on one area. When the guard's attention is fixated (e.g., as determined from a Status attack), the adversary sends contraband through a different approach vector.
- Status: The adversary accesses the guard's orientation and eye movement data following transmission from the HMD to a base station.

To establish confidence that our HMD design will protect the operator and achieve mission objectives, we develop models that combine aspects of human cognitive behavior, HMD hardware and software, and relevant environmental parameters[3]. We then apply formal analysis to prove our design is resilient to the preceding types of attack.

A. Physiological

In this section, we model cognitive vulnerabilities associated with physiological attacks that target the operator's physical well-being through device manipulation. Specifically, we focus on pathways leading to cybersickness induced by adverse hardware behaviors in augmented reality HMDs. To investigate these vulnerabilities, we developed a cognitive model using the Soar cognitive architecture [7] that represents human reactions under conditions that could provoke cybersickness. This model incorporates various HMD hardware and software parameters that could be targeted by a physiological attack, each identified in prior research as critical to inducing physical discomfort: *latency*, and *optic flow*. This model also incorporates users' *exposure time* to cybersickness. While this parameter may not be directly manipulated by an attacker (i.e., an adversary cannot change how long a user has been wearing an HMD), it does still affect the onset of cybersickness and may affect the extent to which a physiological attack is effective. For example, a user who experiences a brief period of high latency after using an HMD for only a few minutes is less likely to be affected by an attack than a user who experiences a period of high latency after wearing an HMD for several hours. With access to personnel logs, an adversary may choose to target HMD operators at the end of their work shift for a more effective attack.

Latency refers to the delay between a user's action (e.g., head movement) and the corresponding update in the virtual element of the MR environment. Latency exceeding acceptable thresholds—typically between 20–30 milliseconds, as recommended by virtual reality (VR) standards—has been shown to cause noticeable user discomfort. Thresholds above 70 milliseconds have been associated with a significant increase in cybersickness symptoms [8]–[10]. The initial values in the physiology model are based on these thresholds from VR research. If needed, these thresholds will be updated in later iterations of the model after further validation activities. Research about latency in AR HMDs is sparse, indicating that latency of virtual objects appearing to users is positively correlated with cybersickness [11] and that latency compensation approaches decrease cybersickness [12], [13]. However, in one AR environment, latency did not affect cybersickness [14]. Thus, the physiology model includes multiple cybersickness parameters to account for the complexity of potential interaction effects.

Optic flow describes the visual motion patterns perceived by users as they or objects within their environment move. Excessive or erratic optic flow, especially when tracking fast-moving objects, has been shown to significantly exacerbate cybersickness, resulting in higher SSQ scores and a greater likelihood of users removing the headset [15]. Research in VR contexts demonstrates that optic flow can affect cybersickness [16], but less is known about its effect on AR. The stable imagery in an AR image may reduce that effect [17] but there is evidence that it is still an important parameter. Higher cybersickness scores are reported in dynamic, moving environments (e.g., tracking moving objects overlaid on a moving starfield) compared with static environments (reading stationary instructions to interact with a real, stationary object) [18], and with virtual objects moving toward the user at higher speeds [19].

Exposure time refers to the amount of time the user has spent wearing the AR HMD. In research involving both VR and AR, cybersickness has been shown to increase with exposure time [17], [20]. However, these effects can be mitigated by taking breaks [17], [21]. In the first iteration of the physiology model, exposure time is defined as short (0-20 minutes) and long (20+ minutes) based on previous studies showing that cybersickness symptoms often develop over the first 20-30 minutes of exposure [18], [20], [22], [23].

These factors were selected for the initial version of the physiology model for three reasons. First, previous studies indicated that they can contribute to cybersickness. Second, data and thresholds can be identified from the literature to use in the model, and third, they can reasonably be the target of an attack (or influence the success of an attack) in the mission scenario. Future iterations of the physiology model may be expanded to include additional factors that contribute to cybersickness and can be targeted in an attack, such as frame rate [12], [24], changing properties of the virtual entities [19], or adjusting the proportion of real to virtual objects in the field of view (i.e., reducing the number of reference objects, or rest frames, in the real world) [22]. However, adding these additional factors will also require more data collection and increase the complexity due to potential interaction effects.

The factors are represented in a state-space model with the ranges defined above where a state exists for every combination of factors. A transition from one state to another is triggered when certain conditions are satisfied (e.g., low latency for 20 minutes leads to a change in cybersickness state from *affected* to *incapacitated*). Some factors are limited in transition, such as exposure time, which only increases, while others are free to transition between all levels of ranges. At any moment a transition may only occur for one factor, rather than multiple simultaneously. Certain states represent hazardous or unsafe states that should be avoided.

We use Soar [7] to represent the physiological preconditions that lead to cybersickness, and the postconditions representing attack effects. For example, a precondition modeled in Soar specifies the latency problem threshold as a trigger for cybersickness and selects the latency-check operator if the current system latency exceeds it. When this operator is applied, the postcondition of the human cognitive behavior is executed, which could be incapacitation. In our Soar model, thresholds for these hardware attributes were incorporated based on

[3]Our models are open-source and publicly available at https://github.com/loonwerks/MATRICS.

values reported in the literature; however, it is important to note that these thresholds have not been formally verified for operational deployment.

A cybersickness score is calculated to determine the user's level of cybersickness. The approach assumes that a user's cybersickness level is a combination of factors occurring over a period of time. Therefore, we represent a user's cybersickness level as the following equation:

$$c_1x_1 + c_2x_2 + c_3x_3 + c_4x_4 \geq [ssq_{low}, ssq_{medium}, ssq_{high}]$$

In an iterative process, this equation determines whether the user will be *not affected*, *affected*, or *incapacitated*. In order to represent exposure time using this AR system, we model the number of times a violation occurs. The maximum time a user spends wearing the AR device will be 20 minutes. Hence, this model measures a maximum of 20 violations for latency and 15 violations for the optical flow factor. Therefore, we use x_1 to represent the total number of times the user is affected by all factors; x_2 represents the total number of times the system latency surpasses the latency lower threshold, x_3 represents the total number of times the system latency surpasses the latency upper threshold, and finally x_4 represents the total number of times the system optical flow surpasses its threshold. c_1, c_2, c_3 and c_4 represent the coefficients of each hardware factor. These coefficients aid in placing the significance of each factor and, in turn, aid in calculating the user's overall cybersickness level. Note that these weights are approximate values and require validation through forthcoming human cognition studies.

The summation of this linear expression is then analyzed using SSQ values, where $ssq_{low} = 10, ssq_{medium} = 20, ssq_{high} > 20$. If the summation is less than or equal to ssq_{low}, the user is not affected; if the summation is greater than ssq_{low} and less than ssq_{medium}, the user is affected. Finally, if the summation is greater than 20, the user is considered incapacitated. Through simulation of these cognitive transitions, we demonstrate the feasibility of integrating models of human physiological responses into formal system analyses.

To validate the behavioral soundness of the cognitive model, we formally verify the Soar-based cybersickness logic by translating it into the nuXmv symbolic model checker [25]. This enables the analysis of dynamic state transitions and operator responses to hardware-induced physiological violations. For instance, one verified property ensures that incapacitating cybersickness does not occur when the latency remains within a safe range:

LTLSPEC G (cl != incapacitated);

where `cl` denotes the operator's cybersickness level. This property ensures that, under normal conditions (e.g., latency between 20-30 milliseconds), the cognitive model does not escalate to severe discomfort. To prevent vacuous proofs and ensure operational relevance, additional properties were verified, covering reachability, safety, liveness, and event responsiveness. These included checks that cybersickness level transitions only occur in response to specific violations (e.g., latency exceeding 70 ms or rapid optic flow), and that the operator remains in valid states and continues progressing toward mission-critical tasks. The Soar model is embedded in a Soar annex (see Fig. 4) of an Architecture Analysis and Design Language (AADL) [26] model and integrated into the Assume-Guarantee Reasoning Environment (AGREE) [27] verification pipeline. Assume-guarantee [28] contracts [29] are specifications for components in a system that state what is expected from the environment in which the component operates and what is required from the component provided the environment meets the assumptions of the contract. Once verified, these behavioral guarantees inform and strengthen the system-level contracts in AGREE. For example, integrating a verified Soar model allows AGREE to prove that the operator will not experience cybersickness under the contract condition that latency stays $\geq$ 70 milliseconds. To resolve remaining contract mismatches, mitigation strategies (such as introducing a latency monitor that dynamically prioritizes rendering tasks) are encoded in the HMD device model, and subsequent AGREE analyses confirms satisfaction of all assumptions and guarantees. This approach demonstrates how cognitive modeling, formal verification, and compositional reasoning can be combined to safeguard human-system interaction against physiological threats.

```
system implementation Operator.impl
    subcomponents
        hardware: system hardware.impl;
        software: system software.impl;

    annex soar {**
        sp {propose*initialize
            (state <s> ^superstate nil
                    ^io.input-link.systemdata <sd>
                    -^name)

        -->
            (<s> ^operator <o> + >,=)
            (<o> ^name initialize)
            (write (crlf) |Agent initializing...|)
        }
```

Fig. 4. Initializing the Soar agent for the Operator in the AADL Soar annex.

B. Perception

In this section, we present an approach to formally capture cognitive vulnerabilities that impact the human perception process. The effects of the perception vulnerabilities are reflected in a task performance model consisting of an operator component and a device component. Using an assume-guarantee reasoning approach, the top-level component, which is the task model derived from the mission scenario, is analyzed for violations of mission guarantees. Specifically, we model the effects of pupil contraction and dilation under a short period of light stimuli in dark ambient conditions, and its impact on the operator's ability to recognize virtual items on the HMD display. The guarantees are non-probabilistic, albeit with an

underlying assumption that it is valid for 95% of the human population.

Cognitive Foundation - In the mission scenario, the task of the operator wearing the HMD is to correctly identify automatic target recognition (ATR) symbols (i.e., bounding boxes). The overall mission requirement is that the human must be able to perform identification at an acceptable level even in the presence of known perception attacks such as a sudden change in luminance. In healthy human subjects, changes in brightness or luminance cause a change in pupil size. Decreased brightness tends to increase pupil size, which enhances light sensitivity and results in a narrower depth of field. Conversely, increased brightness tends to decrease pupil size, resulting in increases in the depth of field and a wider range of clear vision. The contraction and dilation period after a flash of light stimuli is a key parameter in the operator component. We draw upon existing data [30] on pupillometry to create the initial version of the operator model and plan further human-subject experimentation for validation of the operator component.

Cognitive Attack - For perception attacks, we assume the adversary has not compromised the headset system (i.e., they cannot directly manipulate what the headset displays to the user). We assume a bright-light attack from the external environment aimed at the operator to temporarily impair the operator's ability to visually perceive ATR symbols on the HMD display. The adversary has enough control of the external environment to introduce different sources of intense light into the external environment from different locations. The intense light will be targeted at the general direction of the operator, potentially causing compromised visual capabilities.

Mitigation - The mitigation component is a filter capable of dampening or blocking out any intense source of light exceeding a certain threshold above ambient light conditions. We plan to model a variety of filter configurations to illustrate both analysis (i.e., finding cognitive vulnerabilities in the system) and synthesis (i.e., updating the design parameters of the filter to prevent or to reduce the impact of a cognitive attack on task performance).

Formalization - In terms of the underlying formal methods framework, the model is a system architecture containing a collection of AGREE contracts. Each AGREE contract is used to capture either a device component or an operator component behavior. The top-level component captures the overall task of the mission scenario. As shown in Fig. 5, the model consists of contracts composed in a hierarchy where the top-level component is the mission-level contract. This is decomposed into an HMD component and an operator component. The physical environment, mission parameters, and attack are captured using assumptions, while operator performance and mission success are captured using guarantees. The data on pupil dilation and contraction periods are used in the operator component guarantee. Specifically, they are used to compute the amount of delay after a light attack in which the operator's perception capability is severely degraded in the form of $p(\delta_1) \rightarrow q$ where $p(\delta_1)$ is a precondition on the length of time period δ_1 (computed using pupil contraction and dilation data) since a flash of bright light appeared on the HMD display, and q captures the degraded performance of the human operator. The HMD component is further decomposed into subsystems including a mitigation component, which provides localized filtering of any transients (i.e., any sudden increase of light intensity). We created two variations of the model, one with

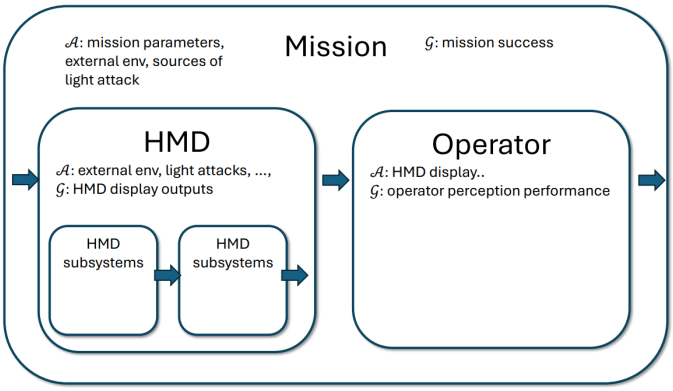

Fig. 5. Illustration of the operator-HMD mission model with assume-guarantee contracts.

a more effective filter subsystem than the other. The less effective filter component has a guarantee that the input light attack is passed through unfiltered to the display. The more effective light filter has a guarantee that it will quickly dampen out any light transients.

Analysis - The AGREE compositional analysis provides a variety of artifacts. For the case where there is a violation of the mission-level contract, the tool returns a counter-example (CEX) showing a trace that invalidates the top-level guarantee. This CEX is returned for the model with the ineffective mitigation component. The CEX is a collection of traces on the inputs and outputs of the contracts of the model. In the context of the mission scenarios, the CEX captures a light attack on the operator. For the variation of the model with the effective mitigation component, the AGREE analysis verifies that the top-level mission contract is satisfied in the model, as anticipated. In this case, there is no CEX because the top-level mission contract is proved to be correct.

Modeling Workflow - The following workflow describes how the perception model is developed.

1) Build AADL model of system including helmet
    a) AADL model composed of components and their interfaces/connections to other components.
    b) AADL components can contain subcomponents.
    c) Each AADL component has a contract (specified in the component's AGREE annex) that guarantees its outputs when assumptions on its inputs are valid.
    d) Contracts for AADL components that are leaf nodes (i.e., do not contain subcomponents), will need to be verified outside of the AADL/AGREE verification framework.

2) Verify that the AADL contracts meet the high-level system requirements.
   a) Compositional verification of AGREE models.
   b) Establish probabilistic contracts on the entire input space vs deterministic contracts on a subset of the input space.
3) Verify contracts (AGREE) for leaf-level AADL components involving human interaction.
   a) The verification of these contracts will depend on satisfying proof obligations on the cognitive models generated from the AGREE contracts.
   b) These proof obligations are to be discharged based on claims derived from cognitive studies, via
      i) Claims from existing literature, or
      ii) From new claims established by experiments.

We plan to refine the operator component further with more features and complexity including different perception modes (auditory and visual), different types of light attacks, multiple sources of attacks, and more complex scenarios with multiple attackers.

C. Attention

Our attention model is a task performance model, with components for the environment, HMD device, HMD operator, and an attacker (see Fig. 6). This model will be used to assess how effectively a user can attend to mission tasks, and to modify or eliminate tasks in order to preserve the successful completion of mission goals. Unlike some of the other models, the attention model is also a *probabilistic* model, demonstrating how MATRICS can model and reason about systems that are non-deterministic. The goal of the analysis is to guarantee that the probability that one or more attackers succeeds in carrying contraband through the checkpoint does not exceed some required bound, K.

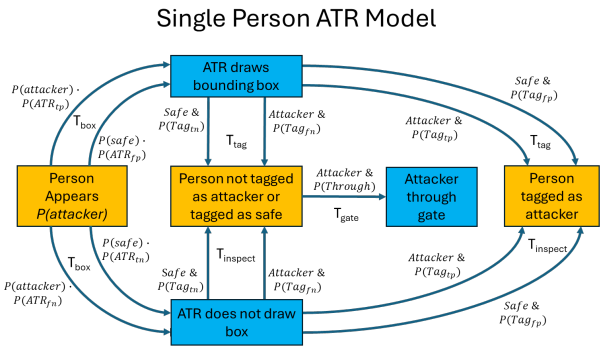

Fig. 6. Attention task performance model for single-person + ATR.

The attention model makes a number of specific assumptions. We assume that the attacker has no access to the headset, but has full control of the external environment. This means that the attacker can introduce some number of people and objects into the environment at any location, subject to some constraints, but the attacker *cannot* directly manipulate what the headset displays to the user.

The model contains elements representing people moving through the environment, some of whom are innocuous, some of whom are attempting to smuggle contraband (attackers), and some of whom are decoys. A person, pi will move towards the checkpoint at a walking speed, and this speed will vary between different persons. It takes person p_i time $T_{CP}(i)$ to reach the checkpoint, where we model $T_{CP}(i)$ as a normal random variable.

For an attacker, a_i, to pass through the checkpoint, the time to reach the checkpoint, $T_{CP}(i)$, must be lower than the time it takes the operator to tag them as unsafe, $T_{tag}(i)$. So, for a single attacker, the desired property is:

$$P\left[T_{CP}(i) < T_{tag}(i)\right] < K$$

There are two cases for tagging an attacker, either: (1) the automatic target recognition (ATR) system in the headset correctly identifies the attacker, and the operator sees the ATR indication and confirms it; or (2) the ATR system does *not* identify the attacker, but the operator identifies them unassisted. Note that if the operator *never* tags the attacker, $T_{tag}(i) = \infty$.

$T_{tag}(i)$ is a function of the false positive and false negative rates of the ATR system in the headset and the user's responses. Depending on the exact technical details of the ATR, its false negative rate can be minimized at the expense of increasing its false positive rate. This has the potential to backfire by increasing the number of bounding boxes the user must attend to, thereby putting more strain on the user's attention.

The user's response is factored into two components. The first is how long the user takes to select a potential attacker (T_{sel}), examine that person (T_{ex}), and then label the person as either a threat or safe (T_{label})[4]:

$$T_{tag}(i) = T_{sel} + T_{ex} + T_{label}$$

Both the time it takes to examine a person, T_{ex}, and the time it takes to use the headset UI to confirm or dismiss a person, T_{label}, are empirical parameters that will be estimated during upcoming human experiments. For now, we conduct formal verification of our attention models across a range of values for each of these parameters to determine likely patterns of responses based on these times.

The second component of the user response is the user's own false positive and false negative rates. The user's false negative rate can be manipulated by instructing them to err on the side of tagging a person as "unsafe" if they are not completely certain the person is "safe." This has the potential to restrict the flow of people through the checkpoint and anger the local population by generating a high number of false positive "unsafe" identifications.

A primary parameter of interest is how long it takes the user to select the next person to examine, T_{sel}. The visual search literature (e.g., [31], [32]) indicates that T_{sel} will be

[4]Again, if any of these processes fail (e.g., the operator never selects a_i), then we treat the delay as infinite (e.g., $T_{sel} = \infty$).

determined to a large extent by the saliency of the target stimuli. In our task domain, saliency can be manipulated using the bounding boxes indicating potential attackers. An object with significantly greater saliency, such as a single bounding box in the display, will be found fairly easily. However, when there are multiple bounding boxes, how the user chooses the next one to examine can be leveraged by an attacker.

We model salience as the relative likelihood that the user will select one bounding box over another. Absent other salience cues, when deciding which among a number of possible objects to direct their attention to, such as which person with a bounding box to examine next, people will generally choose an object that has close spatial proximity to where their attention is already focused (see [33]–[35]). This can be leveraged for a *misdirection* cognitive attack. In this attack, the adversary introduces people to a constrained region of the environment, structuring these introductions to keep the user's attention focused on that region of the environment. Change and inattentional blindness studies [36]–[38] indicate that if an attacker is introduced in a region different from where the user's attention is being focused, there is a high likelihood the user will not notice this new attacker, increasing the probability they reach the checkpoint. This should appear in our model as the new attacker's salience never exceeding that of the distractors already in the area the user is focused on. Thus, mitigating such an attack is a matter of the UI sufficiently increasing the salience of such an attacker, such as by increasing the brightness of a particular bounding box or providing a visual and/or auditory cue directing the user to shift their attention to a different area.

Similarly, in a *flooding* attack, the adversary can introduce a large number of people into the environment designed to trigger false positives from the headset, meaning the user's display will suddenly contain a large number of bounding boxes. If enough bounding boxes are present, the user will be unable to examine all of them before a true attacker reaches the checkpoint. Among the questions we are interested in answering with our human experiments is whether T_{Sel} increases based on the number of bounding boxes present in the environment and, if so, whether that increase is smooth, such as in conjunction search [31], or discontinuous, indicating that cognitive resources have been overwhelmed. Such cases can be mitigated by developing methods that increase $T_{CP}(i)$, thus reducing the time pressure on the user.

Analysis becomes significantly more difficult in cases where there are multiple persons, especially multiple attackers. There are two reasons for this. The first we have discussed already: as more persons are present, the user's attention becomes burdened, so that delays are extended. The second, which we have not yet discussed, is that the guarantee is phrased in terms of *any* attacker crossing the checkpoint. So we can think of each attacker as being a "trial," and the guarantee having to be analyzed against the chance that *all* attackers fail. This is further complicated by the fact that the attackers are not i.i.d. – independent, identically distributed: the chance of one attacker successfully crossing the checkpoint is heavily influenced by the presence of other attackers, and of other travelers. As a result, analytical solutions for determining $T_{CP}(i) < T_{tag}(i)$ for any particular configuration of attackers and distractors become extremely difficult to compute. Modeling the mission, headset, attackers, and user in a probabilistic verification system such as PRISM [39] allows us to run experiments that accurately estimate this probability for a range of values for $T_{CP}(i)$. Based on this, we can then determine ranges for the other variables under our control that will allow our system to satisfy the guarantee.

D. Status

Status is different from the other attack categories because it does not include a cognitive component. Therefore, for this category, we consider traditional cybersecurity mechanisms, specifically focusing on protecting the confidentiality of user biometric information. Formal analysis of our status model uses approaches such as model checking and theorem proving that are already established for formal verification of cyber-resiliency properties in high-assurance systems [40].

In our mission scenario, the HMD operator is a guard in an observation tower. Because the HMD hardware is resource-constrained in terms of power, memory, and processor utilization, collected data are periodically transmitted over an encrypted channel to a dedicated base station for analysis and storage. Although the base station has sufficient resources, it also has a much larger attack surface, providing an easier target for the adversary to hack into.

We therefore model the entire system (i.e., the HMD and the base station) and verify that the confidential operator data cannot be accessed by an adversary in an unauthorized manner. For example, if the adversary had access to gaze data, they would know when it would be safe to operate undetected in a specific area. Our status model, consisting of the HMD, base station, and the communication interface between the two, was developed in AADL (see Fig. 7). Architecture models capture a system's components, interfaces, data flows, and properties, but typically do not describe component behavior. AADL was chosen because it allows engineers to describe the important elements of distributed, real-time, embedded systems (i.e., processors, memory, buses, processes, threads, and data connections) with sufficiently rigorous semantics that can support formal reasoning.

Guaranteeing the confidentiality of operator data against status attacks requires security mechanisms to be part of the system design. In our scenario, we assume it is unlikely that an adversary would be able to directly hack into the HMD itself due to its small attack surface and encrypted communication. We therefore focus on cyber-hardening the base station and verifying confidentiality properties there.

We do so by applying *zero-trust* mechanisms to the system. Zero-trust [41] focuses on moving from a traditional perimeter-based infrastructure, in which the goal is to prevent a breach, to a perimeter-less design, in which a breach is assumed to be likely and the goal is to minimize the effect. This is accomplished by applying zero-trust *tenets* to the system

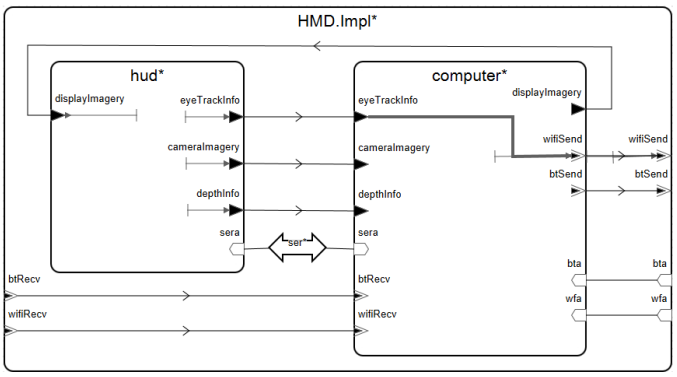

Fig. 7. Top-level HMD system architecture model.

design, which encourage explicit attestation prior to granting access to resources and discourage the use of trust zones.

For this work, we use the BriefCASE framework [42] to add zero-trust components to our system architecture model, guaranteeing confidentiality of operator data. BriefCASE provides a development environment for (1) modeling system architectures in AADL, (2) analyzing the models for cyber-vulnerabilities, (3) mitigating those vulnerabilities by applying automated model transformations, (4) formally verifying security properties in the model, (5) generating high-assurance component code from model specifications, (6) building the system to a secure kernel target, and finally, (7) generating a system cyber-resiliency assurance case. The two key BriefCASE tools that we use to prove status guarantees are AGREE (described above) and Resolute.

Resolute [43] is a language and tool for embedding an assurance argument in an AADL model and evaluating the validity of the associated evidence. Because high-assurance products generally undergo certification at the system level, there is a natural mapping between a system design and the corresponding assurance argument. Resolute takes advantage of this alignment by enabling the specification of the assurance argument directly in the model. The assurance case is then automatically instantiated and evaluated with elements specified in the model. This automated evaluation is possible thanks to the Resolute language and query engine. Queries about the structure or properties of the model can be represented in the Resolute language, which the query engine then interprets and returns a result by traversing the model.

In BriefCASE, Resolute was also utilized to represent cyber requirements. The query engine would then verify the requirements were satisfied in the model. The main benefit of this approach is that requirements that cannot be represented formally can still be specified semi-formally and evaluated against a model (or other development artifacts). In the context of our mission scenario, access to the HMD operator's data must be restricted, otherwise that information could be used maliciously. Therefore, our confidentiality property is "Access to an operator's personal data shall be restricted", which is then refined by lower-level properties: "All messages shall be encrypted prior to transmission" and "Component memory shall be inaccessible by untrusted components". Resolute goals representing these properties are shown in Fig. 8. The goals include the logical rules describing constraints on the architecture that must hold for the guarantees to be satisfied.

```
annex resolute {**
    goal user_data_access_protected(comp : component) <=
        ** "Access to user personal data shall be restricted" **
        no_unencrypted_tx(comp) and no_unauthorized_access(comp)

    goal no_unencrypted_tx(comp : component) <=
        ** "All messages shall be encrypted prior to transmission" **
        forall(conn : connections(comp)) . destination_component(conn) = comp =>
            property(source_component(conn), CASE_Properties::Encrypting) = 100

    goal no_unauthorized_access(comp : component) <=
        ** "Component memory shall be inaccessible by untrusted components" **
        property(comp, CASE_Properties::OS) = "seL4" or
        forall(conn : connections(comp)) . destination_component(conn) = comp =>
            property(source_component(conn), CASE_Properties::Trust_Level) = 100
**};
```

Fig. 8. Resolute goals for assuring confidentiality of operator data.

IV. Conclusion

We are investigating the feasibility of applying formal methods to the cognitive modeling domain to prove guarantees that mixed-reality system operators and missions are protected from cognitive attacks. Our initial results are promising. We have modeled and formally verified HMD application scenarios corresponding to four distinct classes of attack (physiology, perception, attention, and status).

We have several upcoming research activities planned for MATRICS, including:

- Increasing model complexity: Although the fidelity of the models presented in this paper are sufficient for formal reasoning, additional details in both the system and cognitive model components will provide stronger analyses by permitting more general mission scenarios and covering more diverse operator populations.
- Prototyping and validation: In order to demonstrate the feasibility of our approach, we will build MR prototypes and create interactive scenarios that include real-world hardware with embedded applications with which a user can engage. These scenarios will be integrated into novel testing and validation environments, which have a mission scenario at their core, but also integrate systematic parameterization of scenario variables for research purposes and detailed data logging for analysis purposes.
- Assurance: Unlike traditional system cybersecurity, there is a lack of guidance for assuring that MR systems are protected from cognitive attack. Formal analysis results alone will not be sufficient. We are therefore defining *assurance patterns* corresponding to the attack patterns in ReCAP to facilitate evaluation and compliance activities.
- Real-world development: Finally, we look forward to applying our MATRICS framework on real-world MR system development efforts in the aerospace domain.

As mixed-reality systems continue to grow in size and complexity and become more commonplace in aerospace applications, rigorous analysis methods and evaluation criteria must be established (and matured) to provide adequate assurance of protection against adversarial attacks. By addressing this gap

today, MATRICS and other ICS technologies will be ready for adoption before these new attacks become widespread, disruptive, and costly.

V. Acknowledgment

This effort was sponsored by the Defense Advanced Research Projects Agency (DARPA) under agreement number HR0011-24-9-0439. The views, opinions and/or findings expressed are those of the authors and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government.

References

[1] P. Milgram and F. Kishino, "A taxonomy of mixed reality visual displays," *IEICE Trans. Information Systems*, vol. vol. E77-D, no. 12, pp. 1321–1329, 12 1994.

[2] Collins Aerospace, "Helmet Mounted Displays," https://www.collinsaerospace.com/what-we-do/industries/military-and-defense/displays-and-controls/airborne/helmet-mounted-displays, accessed: 2025-05-01.

[3] Varjo, "Varjo XR-4 Secure Edition," https://varjo.com/products/xr-4-secure-edition/, accessed: 2025-05-01.

[4] T. E. Wang and A. Pinto, "Survey of human models for verification of human-machine systems," *CoRR*, vol. abs/2307.15082, 2023.

[5] B. Weyers, J. Bowen, A. J. Dix, and P. A. Palanque, Eds., *The Handbook of Formal Methods in Human-Computer Interaction*. Springer International Publishing, 2017.

[6] MITRE, "CAPEC: Common Attack Pattern Enumeration and Classification," https://capec.mitre.org/, accessed: 2025-05-01.

[7] J. Laird, *The SOAR Cognitive Architecture*. MIT Press, 2012.

[8] G. Papadakis, K. Mania, and E. Koutroulis, "A system to measure, control and minimize end-to-end head tracking latency in immersive simulations," in *Proceedings of the 10th International Conference on Virtual Reality Continuum and Its Applications in Industry*, 2011, pp. 581–584.

[9] A. Kemeny, J.-R. Chardonnet, and F. Colombet, "Getting rid of cybersickness," *Virtual reality, augmented reality, and simulators*, 2020.

[10] J. P. Stauffert, F. Niebling, and M. E. Latoschik, "Latency and cybersickness: Impact, causes, and measures. a review," *Frontiers in Virtual Reality*, vol. 1, p. 582204, 11 2020.

[11] J. Duan, C. Li, G. Yang, C. Qu, E. Chang, Z. Zhang, and X. Che, "Study of cybersickness in augmented reality railway inspections applications," *IEEE Access*, 2024.

[12] J. P. Freiwald, N. Katzakis, and F. Steinicke, "Camera time warp: compensating latency in video see-through head-mounted-displays for reduced cybersickness effects," in *Proceedings of the 24th ACM symposium on virtual reality software and technology*, 2018, pp. 1–7.

[13] T. J. Buker, D. A. Vincenzi, and J. E. Deaton, "The effect of apparent latency on simulator sickness while using a see-through helmet-mounted display: Reducing apparent latency with predictive compensation," *Human Factors*, vol. 54, pp. 235–249, 4 2012.

[14] W. T. Nelson, R. S. Bolia, M. M. Roe, and R. M. Morley, "Assessing simulator sickness in a see-through hmd: Effects of time delay, time on task, and task complexity," in *Image*, 2000.

[15] J. J. Gibson, "The perception of the visual world." 1950.

[16] N. Tian, P. Lopes, and R. Boulic, "A review of cybersickness in head-mounted displays: raising attention to individual susceptibility," *Virtual Reality*, 2022.

[17] C. L. Hughes, C. Fidopiastis, K. M. Stanney, P. S. Bailey, and E. Ruiz, "The psychometrics of cybersickness in augmented reality," *Frontiers in Virtual Reality*, vol. 1, 12 2020.

[18] M. Kaufeld, M. Mundt, S. Forst, and H. Hecht, "Optical see-through augmented reality can induce severe motion sickness," *Displays*, vol. 74, p. 102283, 9 2022.

[19] J. Zhang, X. Che, E. Chang, C. Qu, X. Di, H. Liu, and J. Su, "How different text display patterns affect cybersickness in augmented reality," *Scientific Reports*, vol. 14, no. 1, p. 11693, 2024.

[20] R. E. Haamer, N. Mikhailava, V. Podliesnova, R. Saremat, T. Lusmägi, A. Petrinec, and G. Anbarjafari, "Motion sickness in mixed-reality situational awareness system," *Applied Sciences*, vol. 14, p. 2231, 3 2024.

[21] T. A. Doty, J. W. Kelly, S. B. Gilbert, and M. C. Dorneich, "Cybersickness abatement from repeated exposure to vr with reduced discomfort," *IEEE Transactions on Visualization and Computer Graphics*, 2024.

[22] R. Kirollos and W. Merchant, "Comparing cybersickness in virtual reality and mixed reality head-mounted displays," *Frontiers in Virtual Reality*, vol. 4, p. 1130864, 2 2023.

[23] K. A. Pettijohn, C. Peltier, J. R. Lukos, J. N. Norris, and A. T. Biggs, "Virtual and augmented reality in a simulated naval engagement: Preliminary comparisons of simulator sickness and human performance," *Applied Ergonomics*, vol. 89, p. 103200, 11 2020.

[24] J. Wang, R. Shi, W. Zheng, W. Xie, D. Kao, and H.-N. Liang, "Effect of frame rate on user experience, performance, and simulator sickness in virtual reality," *IEEE Transactions on Visualization and Computer Graphics*, vol. 29, no. 5, pp. 2478–2488, 2023.

[25] NuSMV Model Checker, http://nusmv.itc.it.

[26] P. H. Feiler and D. P. Gluch, *Model-Based Engineering with AADL: An Introduction to the SAE Architecture Analysis and Design Language*, 1st ed. Addison-Wesley Professional, 2012.

[27] M. W. Whalen, A. Gacek, D. Cofer, A. Murugesan, M. P. Heimdahl, and S. Rayadurgam, "Your "what" is my "how": Iteration and hierarchy in system design," *IEEE Software*, vol. 30, no. 2, pp. 54–60, 2013.

[28] T. A. Henzinger, S. Qadeer, and S. K. Rajamani, "You assume, we guarantee: Methodology and case studies," in *Computer Aided Verification: 10th International Conference, CAV'98 Vancouver, BC, Canada, June 28–July 2, 1998 Proceedings 10*. Springer, 1998, pp. 440–451.

[29] A. Benveniste, B. Caillaud, D. Nickovic, R. Passerone, J.-B. Raclet, P. Reinkemeier, A. Sangiovanni-Vincentelli, W. Damm, T. A. Henzinger, K. G. Larsen *et al.*, "Contracts for system design," *Foundations and Trends in Electronic Design Automation*, vol. 12, no. 2-3, pp. 124–400, 2018.

[30] K. Tekin, M. A. Sekeroglu, H. Kiziltoprak, S. Doguizi, M. Inanc, and P. Yilmazbas, "Static and dynamic pupillometry data of healthy individuals," *Clinical and Experimental Optometry*, vol. 101, no. 5, pp. 659–665, 2018.

[31] A. Treisman, "Perceptual grouping and attention in visual search for features and for objects." *Journal of experimental psychology: human perception and performance*, vol. 8, no. 2, p. 194, 1982.

[32] H. J. Müller and J. Krummenacher, "Visual search and selective attention," *Visual Cognition*, vol. 14, no. 4-8, pp. 389–410, 2006.

[33] Z. Chen, "Object-based attention: A tutorial review," *Attention, Perception, & Psychophysics*, vol. 74, pp. 784–802, 2012.

[34] G. D. Logan, "The code theory of visual attention: an integration of space-based and object-based attention." *Psychological review*, vol. 103, no. 4, p. 603, 1996.

[35] E. Awh, K. M. Armstrong, and T. Moore, "Visual and oculomotor selection: links, causes and implications for spatial attention," *Trends in cognitive sciences*, vol. 10, no. 3, pp. 124–130, 2006.

[36] A. Mack, "Inattentional blindness: Looking without seeing," *Current directions in psychological science*, vol. 12, no. 5, pp. 180–184, 2003.

[37] M. S. Jensen, R. Yao, W. N. Street, and D. J. Simons, "Change blindness and inattentional blindness," *Wiley Interdisciplinary Reviews: Cognitive Science*, vol. 2, no. 5, pp. 529–546, 2011.

[38] D. J. Simons and C. F. Chabris, "Gorillas in our midst: Sustained inattentional blindness for dynamic events," *perception*, vol. 28, no. 9, pp. 1059–1074, 1999.

[39] A. Hinton, M. Kwiatkowska, G. Norman, and D. Parker, "Prism: a tool for automatic verification of probabilistic systems," in *Proceedings of the 12th International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, ser. TACAS'06. Berlin, Heidelberg: Springer-Verlag, 2006, p. 441–444.

[40] D. Cofer, A. Gacek, J. Backes, M. W. Whalen, L. Pike, A. Foltzer, M. Podhradsky, G. Klein, I. Kuz, J. Andronick, G. Heiser, and D. Stuart, "A formal approach to constructing secure air vehicle software," *Computer*, vol. 51, no. 11, pp. 14–23, 2018.

[41] S. Rose, O. Borchert, S. Mitchell, and S. Connelly, "Nist special publication 800–207 zero trust architecture," *National Institute of Standards and Technology, US Department of Commerce*, pp. 800–207, 2020.

[42] D. Cofer, I. Amundson, J. Babar, D. Hardin, K. Slind, P. Alexander, J. Hatcliff, Robby, G. Klein, C. Lewis, E. Mercer, and J. Shackleton, "Cyberassured systems engineering at scale," *IEEE Security & Privacy*, vol. 20, no. 03, pp. 52–64, May 2022.

[43] A. Gacek, J. Backes, D. Cofer, K. Slind, and M. Whalen, "Resolute: An assurance case language for architecture models," in *HILT 2014*. New York, NY, USA: ACM, 2014, pp. 19–28.