THE UNIVERSITY OF ALABAMA IN HUNTSVILLE

Navigating Uncertainty: Enhancing AI System Safety through the Integration of Systems Theory, Set Theory, and R3+ Concepts

Presentation for 2024 AI for Systems Engineering and Systems Engineering for AI Conference

Reginald Holmes, MEng, CSEP, PMP

Ph.D. Student (Systems Engineering) - The University of Alabama in Huntsville

Sr. Principal Systems Engineer - Northrop Grumman

Dr. Hanumanthrao Kannan

Assistant Professor - The University of Alabama in Huntsville

"Human interference caused a dozen Cruise robotaxis to stall, creating a city-wide traffic jam and highlighting the vulnerability of AI systems to unpredictable human behavior in urban environments."

Key Points from the Cruise Robotaxi incident

- **Human Interference:** The incident revealed how autonomous systems can be easily disrupted by intentional human actions, underscoring a critical safety vulnerability.
- **Public Behavior's Impact:** The event showed how individual behaviors can compromise AI safety, affecting not just passengers but also other road users.
- **Urban Challenges:** The incident highlighted the complexities autonomous vehicles face in urban environments, especially when dealing with unpredictable human interactions.
- **Need for Better Safety Protocols:** It underscored the necessity for robust safety frameworks to manage unexpected disruptions without causing hazards.

Defining AI System Safety Challenges

Challenge	Details	Systems Engineering Insight	Definitions & Clarifications
Defining the Problem Space	Defining safety within the problem space involves scenario analysis, stakeholder mapping, and capturing dynamic interactions between AI systems and their environments.	A precise and formal problem space is crucial for design and development in systems engineering. This ensures alignment with stakeholder expectations and safety goals..	**Problem Space:** A structured representation of the needs, requirements, and desired outcomes necessary to define the scope and constraints of the system solution.
Limitations of Requirements Engineering	Traditional methods struggle with scaling to complex AI systems, leading to issues in unambiguity, traceability, verifiability, consistency, and completeness.	A unified theoretical foundation based on formal method such as Set Theory is needed to ensure that requirements are clear, traceable, testable, and consistent across complex systems. These formalisms address inherent limitations in heuristic-based approaches.	**Unambiguity:** Requirements must be clear and unambiguous to avoid misinterpretation. **Traceability:** Ability to trace requirements back to stakeholder needs.
Completeness and Hazard Analysis	Insufficient hazard analysis often stems from incomplete problem space definitions, leading to gaps in risk identification and mitigation.	Ensuring completeness requires the analysis of all relevant safety risks and their alignment with stakeholder needs. Formal methods such as Systems Theory ensure that every aspect of the problem space is captured, evaluated, and verified for completeness.	**Completeness:** Ensures that all necessary aspects of the system, including hazards, are captured. **Consistency:** All requirements must align without contradictions.
Capturing Safety Requirements	Misunderstanding safety or conflating it with uncertainty leads to unclear or incomplete safety requirements. There is a need for clear definitions of safety in AI systems.	A formal approach using Systems Theory and Set Theory is critical for defining and applying safety concepts in AI design. The structured frameworks ensure clarity, consistency, and the application of formal theorems to analyze the correctness of safety requirements.	**Verifiability:** Ability to test and confirm that requirements are met. **Consistency:** Requirements must not conflict with each other, maintaining logical coherence.

Boeing 737 MAX and the MCAS System: Lessons from Poor Problem Space Definition

In 2017, Boeing introduced the MCAS (Maneuvering Characteristics Augmentation System) software as part of the 737 MAX to address a specific aerodynamic challenge. However, this case exemplifies how an inadequate definition of the problem space can lead to significant safety issues.

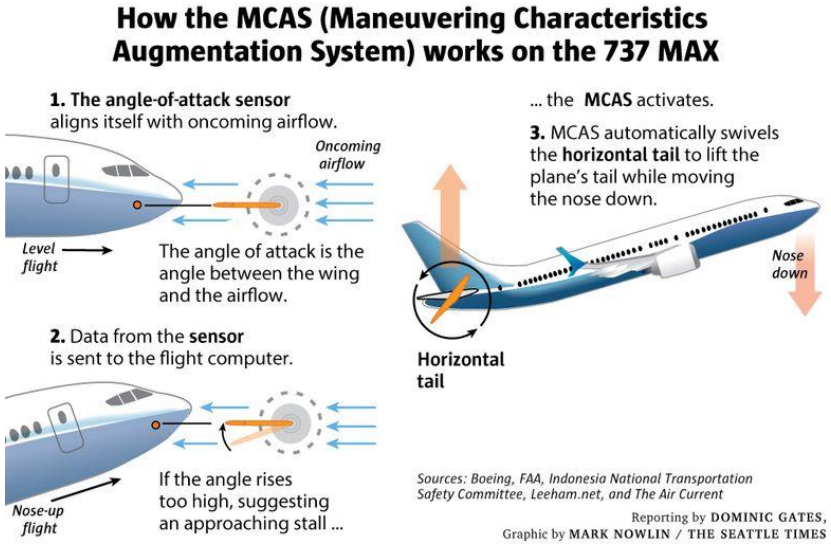
Key Issues:

Larger Engines: The 737 MAX had larger, more fuel-efficient engines mounted higher and further forward, altering the aircraft's aerodynamics and causing a pitch-up tendency.

Aerodynamic Challenge: This new engine placement led to a nose-up pitch at high angles of attack, increasing the risk of stalling.

MCAS Introduction: The MCAS was designed to automatically adjust the stabilizer to correct the pitch-up tendency, making the handling of the 737 MAX similar to earlier models.

Minimal Retraining: Boeing opted to minimize retraining by keeping the 737 MAX's handling similar to earlier models, despite the significant aerodynamic changes.



Defining AI System Safety Challenges in the Context of Boeing

Poor Definition of the Problem Space:

- Boeing identified the aerodynamic challenge (pitch-up tendency) but did not fully understand or address the broader implications and risks of introducing the MCAS system.
- Consequences: This lack of a comprehensive problem space led to a failure in capturing all safety hazards associated with MCAS, which ultimately resulted in the fatal crashes

Completeness and Hazard Analysis:

- Insufficient consideration of how MCAS would interact with real-world flight conditions and pilot responses led to incomplete hazard identification.

Limitations in Requirements Engineering:

- The system was designed with an inadequate safety framework, lacking clarity, traceability, and verifiability in its requirements for safety-critical scenarios.
- There was ambiguity in how MCAS operated without pilot awareness, leading to inconsistent safety outcomes.

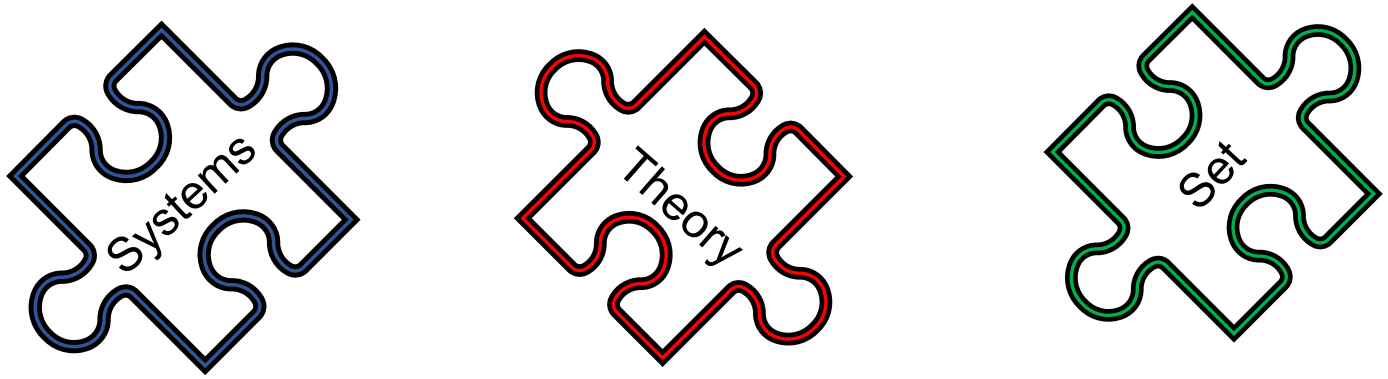
Capturing Safety Requirements:

- Boeing misunderstood the implications of introducing a new system (MCAS) and its effects on overall safety, failing to clearly define safety in the problem space, a core challenge in AI system safety.

"The flaws in the software design that took flight control away from the pilots without their knowledge based on data from a single sensor ultimately led to the two 737 MAX crashes in 2018 and 2019, causing the deaths of 346 people." — Bill George, Harvard Business School – Working Knowledge

Systems Theory + Set Theory

- **Framing with Systems Theory:** Provides the conceptual foundation to identify key problem space elements (functions, outcomes, requirements, needs) and their interactions. However, it lacks the formalism to ensure consistency, completeness, and traceability.
- **Formalizing with Set Theory:** Adds the mathematical rigor needed to define sets, functions, and relationships, allowing us to formally express and analyze the problem space structure.
- **Addressing Ambiguity & Inconsistency:** Set Theory ensures that all elements are mathematically modeled and relationships are clearly defined, avoiding ambiguity and inconsistency.
- **Ensuring Traceability & Verifiability:** Set Theory maps system requirements back to stakeholder needs, enabling traceability and verification within the system model.
- **Achieving Completeness & Consistency:** Set Theory guarantees a complete and consistent problem space by formally defining all relevant elements and ensuring alignment without contradiction.



Defining Safety and R3+ using Systems Theory + Set Theory

Define Safety Conceptually:

- Use Systems Theory to define safety within the system, identifying key elements like functions, outcomes, and requirements.

Formalize with Set Theory:

- Apply Set Theory to provide mathematical rigor, formalizing safety definitions and integrating R3+ concepts (Robustness, Reliability, Resilience, Antifragility) into the system design.

Build and Align the Safety Model:

- Combine Systems Theory and Set Theory to construct a formal safety model that captures, relates, and ensures the alignment and consistency of all safety and R3+ elements.

Distinguish Safety from R3+:

- Use this process to clearly differentiate safety from R3+ concepts, enabling a deeper understanding of whether R3+ equates to safety or if they are distinct.

Mapping R3+ and Safety Concepts to Systems Theory Elements

R3+ / Safety Concept	Related Systems Theory Element	Role & Specificity
Robustness	Functions f, Operational Context $OpsCtx$, System Solution $SysSol$	**Role**: Robustness is about maintaining function performance despite changes in the operational context. **Specificity**: Ensures that functions $f(i(t), C)$ produce the expected output $o(t)$ consistently, even when input trajectories $i(t)$ and conditions C vary within the operational context $OpsCtx$.
Resilience	States SZ	**Role**: Resilience focuses on the system's ability to recover from disruptions and transition into new operational states. **Specificity**: Evaluates how quickly the system can return to a stable state SZ' after a disturbance, ensuring continued operation of functions f.
Reliability	Input/Output Trajectories $i(t), o(t)$	**Role**: Reliability depends on consistent performance of system functions over time. **Specificity**: Defines how input trajectories $i(t)$ consistently lead to correct output trajectories $o(t)$, maintaining system functionality.
Antifragility	System Solution $SysSol$	**Role**: Antifragility concerns the system's ability to improve through challenges. **Specificity**: Defines how the system solution $SysSol$ adapts and strengthens in response to stressors, evolving to a more robust state after exposure to challenges.
Safety	Outcomes Oc	**Role**: Safety ensures that system operations do not lead to harmful outcomes. **Specificity**: Tied to the definition of desired outcomes Oc, ensuring no harmful consequences occur as the system interacts with external systems E within the operational context $OpsCtx$.

Formal Definitions – Robustness

Robustness R is a key attribute that applies differently depending on whether a system operates within a **closed or open system context**.

- In a **closed system**, where the system solution $SysSol$ does not interact with external systems $(E = \emptyset)$ robustness is primarily linked to the system's individual functions f, ensuring that they maintain operational integrity despite variations or disturbances in input trajectories $i(t)$, internal states SZ, or specific conditions C. Here, robustness ensures that the functions themselves remain stable and continue to deliver consistent output trajectories $o(t)$.
- However, in an **open system** context, where the system solution $SysSol$ interacts with external systems E within the operational context $OpsCtx$ $(O \cap E \neq \emptyset)$, robustness extends beyond individual functions and relates to the system solution $SysSol$ as a whole. It ensures that not only do the individual functions maintain performance, but also that the interactions between the system and external systems E within the operational context $OpsCtx$ allow the entire system solution to maintain its operational goals and meet desired outcomes Ocd , even in the face of environmental disturbances and interactions with external elements.

- This definition of robustness is derived from a thorough literature review, integrating insights from the various works listed (Kapur & Reed, 2014; Kitano, 2007; Clément et al., 2021; Carlson & Doyle, 2002) that discuss robustness in the context of systems theory and engineering. The selected definition specifically aligns robustness with key systems theory elements—such as functions, outcomes, and system solutions—making it particularly appropriate for analyzing complex systems where interactions between components are crucial.

- **High Level Goal:** The primary goal is to ensure **system safety**, with robustness as a key factor. Robustness thus ensures both functions and the system as a whole can handle disturbances while achieving safe outcomes.

The journal paper will provide formal definitions of Reliability, Resilience, and Antifragility later since they are not formally defined yet.

Taxi Example: Understanding Robustness in Context

Cruise Robotaxi Incident: In the context of the Cruise Robotaxi incident, robustness would mean that even when faced with human interference or unexpected behavior in the urban environment, the robotaxi should maintain its operational integrity, preventing stalls and ensuring safe and smooth transportation.

Functions in Context:
Navigation Function $fnav$:

- The $fnav$ is responsible for guiding the robotaxi along its intended route. This includes input trajectory $i(t)$— such as GPS coordinates, road conditions, and traffic signals to determine the vehicle's path.
- Robustness Example: If a pedestrian suddenly steps in front of the vehicle, the navigation function must adapt its output trajectory $o(t)$ without deviating from the overall route plan. Robustness here ensures that the vehicle avoids obstacles while staying on course.

Obstacle Detection Function $fobs$: This $fobs$ processes inputs from sensors to detect obstacles in real time.

- Robustness would mean that even if external conditions C change suddenly (e.g., a person suddenly obstructing the vehicle's path), the $fobs$ must accurately identify and react to these new inputs $i(t)$without failing or producing false positives, ensuring the vehicle remains safe.

Outcomes in the Cruise Robotaxi Context:
Desired Outcomes Oc_d:

- The main outcome expected from the Cruise Robotaxi is to transport passengers safely and efficiently from one point to another within an urban environment.
- Robustness in Outcomes: Robustness at the outcome level means that, despite any disturbances (like human interference or sudden traffic changes), the robotaxi system as a whole continues to achieve its desired outcome without causing accidents or delays.

Formal Definitions – Safety

- Safety Saf is defined as the condition where a system solution $SysSol$ operates within its intended operational context $OpsCtx$ without causing harm or leading to unacceptable losses. Unacceptable losses are defined as outcomes Oc that significantly deviate from stakeholder safety criteria, which are established based on the severity and probability of potential adverse events. Safety is achieved not only through enhancing component reliability but more critically, by systematically identifying, managing, and mitigating risks to prevent hazards. Hazards are specific conditions or malfunctions that could lead to critical system failures or accidents. (Leveson, 2012; Saleh & Marais 2005).
    - The integration of **Leveson's** and **Saleh & Marais'** safety definitions ties back into **Systems Theory** by framing safety as a system-wide property that emerges from the interactions between components, their behaviors, and the external environment.

- **Distinguishing from R3+ Concepts:** Distinguishing from R3+ concepts (robustness, reliability, and resilience), which enhance the system's inherent stability and response capabilities, safety encompasses the design and operational practices that continuously align system states SZ and outcomes Oc with safety objectives. This proactive approach minimizes the likelihood of scenarios leading to harmful consequences, emphasizing that high component reliability alone does not equate to system safety.

- **High-Level Goal:** Safety ensures that all system operations strictly adhere to defined safety criteria under any conditions to prevent harm. It focuses on preventing hazards and minimizing the likelihood of adverse events by addressing both the severity and probability of risks. This clear focus on risk mitigation and accident prevention makes safety distinct from robustness, resilience, and reliability, which concentrate on system performance and recovery.

Path Forward

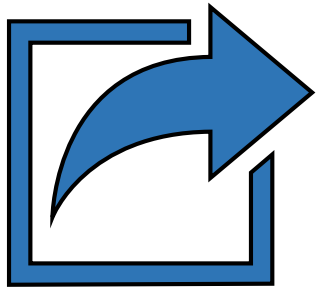
Key Insights:

- **Integration of Systems and Set Theory:** The presentation emphasized the critical role of integrating Systems Theory and Set Theory to create a precise and rigorous foundation for AI system design. This approach ensures that every element within the system is clearly defined and consistently applied across the design and operational phases.

- **R3+ Concepts for Safety:** Robustness, reliability, resilience, and antifragility (R3+) were identified as essential attributes that contribute to the safety and stability of AI systems. These concepts, when quantitatively defined and applied, enhance the system's ability to handle uncertainties and operate within safe limits.

The Path Forward:

- **Further Development of Formal Definitions for R3+**
- **Develop a Fundamental Safety Metric**
- **Expand and Refine Axioms and Theorems**
- **Broader Application of Formal Language**
- **Integration of Interdisciplinary Insights**



Questions