Table of Contents

Appendix C Extended Literature Review

C.1 General Systems Science Concepts

C.1.1 Complexity and Complicatedness

Reviewing the literature on complexity and complicatedness [1], the traditional systems engineering paradigm appears to have a vague distinction between complexity and complicatedness. The INCOSE [2] definition of complexity distinguishes between **simple, complicated, and complex systems** based on the nature of their elements and the relationships between those elements:

1. **Simple Systems**: Relationships between elements are easily observed and comprehended, leading to straightforward cause-and-effect understanding.
2. **Complicated Systems**: Relationships are intricate but can be unravelled and comprehended, enabling sufficient certainty about cause and effect. Traditional systems engineering (SE) methods work well here.
3. **Complex Systems**: Relationships between elements are woven together and not fully comprehended, leading to uncertainty about cause and effect. These systems often exhibit emergent behaviours, nonlinear dynamics, and counterintuitive outcomes.

As a holistic systems engineer, how do I use this definition objectively? There is no direct relationship to an already quantifiable aspect that can be re-adapted to solve real-world problems. Yes, there is an element of appreciation, but it is hard to translate such a definition in practice. This is because the definition does not tell me what a not-system is. INCOSE's definition tells me that all predictable organisations and unpredictable organisations of things are systems. But, I need a clear boundary of 1 and 0 so that I can say this is a bad choice because it leads to no system, and this is a good choice because it leads to the realisation of a system.

INCOSE does not directly define complicatedness and how it differs from complexity. So, based on INCOSE's explanation of complicated systems, **complicatedness** can be inferred as:

Complicatedness refers to the quality of a system having intricate but comprehensible relationships between its elements. These relationships can be systematically analysed, understood, and managed, leading to **sufficient certainty** in predicting cause-and-effect dynamics.

- **Key Characteristics of Complicatedness according to INCOSE**:

1. **Detailed Interconnectivity**: While the relationships among elements are numerous or detailed, they are not inherently unpredictable or emergent.
2. **Deterministic or Predictable Behaviour**: The system behaves according to fixed or known rules and patterns, allowing its behaviour to be unfolded and understood through analysis.
3. **Amenability to Reductionist Methods**: Problems can be broken down into smaller parts, solved individually, and recombined to form the overall solution.
4. **Finite State Space**: The system has a bounded and comprehensible range of possible states or outcomes.

Example: A **complicated system** like a traditional car engine has many interconnected parts (e.g., pistons, valves, fuel injectors), and their interactions can be intricate. However, engineers can understand, predict, and manage these interactions using well-established principles and methods.

According to the INCOSE characterisation of Simple and Complex, why is a predictable car engine not simple? Why is it not Complex? Would an autonomous car be a complex or complicated system? What would you class a domino puzzle of 100 pieces and 10^9 pieces?
In summary, INCOSE appears to draw the following distinction:

Complicatedness:

- Defined by manageable intricacy.
- Relationships are fixed and predictable.
- Reductionist approaches (e.g., decomposition, reassembly) work well.

Complexity:

- Defined by emergent behaviours.
- Relationships are interdependent and non-linear.
- Requires holistic, iterative, and systems thinking approaches.

At the same time, INCOSE distinguishes the following properties of emergence in general:

- **Emergence is a System-wide Property**: Systems exhibit emergent properties that derive from their elements and interactions but cannot be reduced to those elements [3].
- **Universal Phenomenon**: Emergence is recognised as a fundamental property of all systems [4].
- **Distinct from Elemental Properties**: Emergence arises from the "properties the system has, but the elements themselves do not" [5].

Interestingly, the cited reference in [4], Sillito et.al, postulate the following:

"Emergence, the appearance of a new phenomenon or capability as a result of relation or interaction between objects, is key in differentiating between objects that are systems and those that are not."

In their remarks, they implicitly state that objects that do not exhibit emergent behaviours are not systems. It is interesting that the authors consider the feasibility of "non-system" to exist. So, according to INCOSE, complicated systems do not exhibit emergent properties. And according to Sillito et al., complicated systems are not systems. This is the problem we found in the literature, which is confusion and ambiguity.

So what? Why is understanding the difference between complexity and complicatedness important for our systems approach and for any aspiring systems engineer working on safety-critical autonomous systems?

Simply because Compounded Epistemic Uncertainty Problems (problems that involve autonomous systems, people, traditionally engineered systems, environment, etc.) in a safety-critical application are confusingly complicated issues and hard to achieve robust certainty (robust trustworthy safety case) that can withstand the test of time, in comparison to the issues that involve engineering deterministic safety-critical CPS. Understanding the nature of complicatedness is the key aspect of reducing epistemic uncertainty for Compounded Epistemic Uncertainty Problems.

This section is covered in more detail in our conference paper, which outlines the core concepts of complexity and complicatedness in the following work [6]. We will briefly explain our prior work as the foundation for the

C.2 Deeper articulation of research problems

C.2.1 Research problem definition

Initially, we set out to examine potential concerns to understand the scope of the problem domain. The concerns were mainly related to the following preconceptions:

1. There is no clear approach to developing a comprehensive training process for machine learning models by designing training data. To our understanding, no universally adopted or standardised process exists for designing and ensuring training data for machine learning models.

2. There is no clear methodology for reducing architect uncertainty about the dataset's trustworthiness, arising from the semantic gap between desired emergent intelligent functionality and pictorial data sets.

Throughout the PhD project, the problem was revisited several times, and we redefined the issues in terms of risks that may be encountered during the design of autonomous systems or challenges that are uniquely difficult for autonomous-systems engineering (ase) processes:

C.2.2 ASE problem 1

The risk of solving the wrong or incomplete problem due to the architect's epistemic uncertainty and biased perception of the problem complexity dynamics

So, what do we mean by solving the wrong or an incomplete problem? Let's explain it in a hypothetical example scenario:

Initial assumption – the "wrong" problem

An autonomous vehicle architect is tasked with designing a vision-based pedestrian detection system for urban environments. The initial assumption is that occlusion and poor lighting are the primary challenges affecting pedestrian detection accuracy. Based on this, the team spends six months developing an advanced sensor fusion approach, integrating thermal cameras and lidar to improve pedestrian recognition in low-light and occluded scenarios.

Discovery of the "real" problem – six months later

During real-world testing, engineers notice that the pedestrian detection system fails in unexpected scenarios, even in well-lit conditions with no occlusions. After extensive debugging, they realise that the real issue is not sensor limitations but adversarial interference:

- The detection system struggles with false negatives when pedestrians ***wear clothing patterns*** that confuse the AI model (e.g., dressing up for public holidays).
- The system misclassifies children and small individuals due to biases in the training dataset, which was heavily skewed toward adult pedestrians.
- Specific reflection patterns from glass buildings create phantom pedestrians, causing unnecessary emergency stops.
- Statues and pictures of people on buses billboards fool the perception system to detect people nearby by thus taking a drastic action in the middle of the road (as intended).

Consequences – 5 months of work need to be reworked

The team must rework five months of effort (including updating the hazards analysis and introducing new missing requirements that were not considered or thought of during the contract evaluation and agreement), because their iterative solution was designed around sensor fusion improvements rather than addressing adversarial perception risks and dataset biases.

For example, whenever they see that the system keeps getting false negatives, they assume that the problem is ***"not enough depth in sensing"***, prompting them to add different ways to improve or support sensing. For example, initially, they rely on rgb cameras only, but then the system fails to detect pedestrians sufficiently, so they solve it by adding lidar. Again, the perception system kept getting more false negatives, so they included infrared. Less they know the issue is ***"epistemic uncertainty"*** problem, rather than sensor problem. This is an example of how computational thinking can let down a group of highly talented team.

a new architect who is a systems thinker is employed to understand why. Contrary to common belief, the architect does not inspect the system but the environment and realises that the real issue is the list above.

Now, the new problem requires:

- Redesigning the perception model to handle adversarial examples and Black Swan scenarios.
- Revisiting training datasets requirements to ensure diversity in pedestrian appearances, body sizes, and environments.
- Re-work the entire hazards analysis and re-produce HAZOP reports. The team may realise that using HAZOP was not enough; now, they need a more holistic approach to ensure that they face fewer surprises.

This mistake delayed deployment by an additional three months and significantly increased project costs. All because the architect's initial epistemic uncertainty led them to solve the wrong problem—sensor limitations instead of AI bias and adversarial vulnerabilities.

Lesson learned

This example highlights the critical risk of architects' epistemic uncertainty in autonomous system design. Relying on surface-level problem assumptions without deep validation can lead to months of wasted effort and massive redesigns. A more iterative problem-validation approach—such as progressive real-world testing and adversarial scenario simulations during early design stages—could have avoided this costly mistake.

If the design process underestimates the complexity of the problem domain, it may lead to oversimplified solutions that overlook critical interdependencies, emergent risks, and unforeseen scenarios. This oversight can compromise the system's reliability, safety, and effectiveness in real-world applications. The tendency to underestimate perceived complexity is closely linked to the architect's predictive thinking process and their level of situational awareness regarding the problem. This awareness is essential for making well-informed engineering decisions and addressing the situation's complexity in a manner that makes the resulting design decisions sufficiently trustworthy.

The risk of making uninformed engineering predictions can lead to misaligned system requirements. This often occurs due to biased or incomplete understanding [7], stemming from incorrect or missing assumptions and insufficient thought processes. Such misalignment can negatively impact various stakeholders, including problem holders, the architecture team, and technical delivery teams. Cognitive biases and uncertainty of autonomous systems and the open operational environment’s emergent behaviours pose a complex and complicated problem that requires a rigorous approach to predict what will happen. The example of the food delivery robot contextualises the risk that system engineers and designers may fail to adequately comprehend autonomous systems' complex, emergent behaviours in their operating environments.

The effectiveness of any systems engineering approach for autonomous systems is bounded by its ability to enable the architect to predict the complexity of the problem domain as-is and to-be after deploying the autonomous solution. FMEA [8], HAZOP [9], and STAMP [8], in our experience, do not provide the intellectual rigour that would enable designers to effectively resolve the complicatedness of autonomous systems problem domains by thoroughly exposing the hidden Black Swan scenarios and thus making it somewhat predictable.

Without a thorough understanding of the problem domain and its contextual factors, emergent properties, those that arise unexpectedly during operation, may be overlooked. This can lead to incomplete or inaccurate high-level system requirements, misaligning the intended system capabilities with its actual performance. This could lead to unforeseen failures, inefficient system design, and suboptimal user experience in dynamic and complex environments.

We believe this risk is directly linked to human cognitive performance under uncertainty. Tversky and Kahneman [7] demonstrated that people often rely on heuristic principles—mental shortcuts—to decide about uncertain events. While these heuristics simplify complex assessments of probability and value, they frequently lead to systematic errors and biases.

The study highlights three main heuristics:

- **Representativeness heuristic**: People judge the probability of an event based on how similar it is to their existing prototype of such events, neglecting statistical realities such as base rates or sample size.

- **Availability heuristic**: The likelihood of events is estimated based on how easily examples come to mind. This can distort judgment since vivid or recent events may be disproportionately weighed.
- **Anchoring and adjustment heuristic**: When estimating values, individuals tend to be influenced by an initial anchor (a reference point) and make insufficient adjustments from that anchor, even when the anchor is arbitrary.

Despite their utility, these heuristics often lead to predictable biases:

- Overconfidence in predicting events based on small or non-representative experiences.
- Misjudging probabilities, such as neglecting the law of large numbers or regression toward the mean.
- Failure to internalise basic statistical principles, even when individuals encounter relevant examples throughout their lives.

The study also emphasised that these biases occur not only in people but also in inexperienced researchers, including those trained in statistics, suggesting that intuitive, subjective judgment remains vulnerable to these systematic errors, regardless of experience.

These insights about human cognitive biases are profoundly relevant in the context of engineering judgment for intelligent system design and architecture, particularly in uncertain and complex environments. When designing solutions for open, harder-to-predict systems, especially when dealing with Multiple interacting autonomous systems, engineers face similar challenges in making decisions under uncertainty. The risk of making mis-engineering judgments is higher than that of making regular daily life decisions.

Key parallels include:

- **Relying on heuristics in complex environments**: just as humans simplify probabilistic reasoning using heuristics, deterministic systems architects might rely on simplified models or assumptions to handle the uncertainty in autonomous systems' behaviour and environment, for example. However, such an approach can lead to underestimation of risks or failure to account for rare, high-impact Black Swan events [10], much like human biases when using availability or representativeness heuristics.
- **Overconfidence and anchoring**: problem stakeholders and engineers might anchor on initial design assumptions [2] or specific performance metrics, adjusting insufficiently as more data about system behaviour in real-world, complex environments becomes available. This is like the anchoring and adjustment bias observed in human judgment [11].

- **Overlooking statistical realities**: engineers may fail to fully account for intelligent systems' inherent unpredictability and variability, especially when Multiple agents or machines interact. Assuming that, for instance, if the information is in the dataset, the model shall always detect it. Unaware of the fact, just because we have 100,000 images of a drone to train a model, and a testing result shows the model will 99% detect a drone, in the real world, 99% of the instances will be detected. This is a deterministic system engineering mindset. The failure to grasp statistical concepts like regression to the mean or the effect of sample size, as seen in human judgment, can lead to misguided confidence in the robustness or accuracy of these systems.

In summary, Tversky and Kahneman's seminal work demonstrated that human judgment under uncertainty [7] is systematically prone to biases such as overconfidence, anchoring, and the availability heuristic. These findings underscore the inherent challenges that even sophisticated cognitive systems like the human brain face when navigating complex and unpredictable scenarios. By analogy, autonomous systems, which humans design, inherit similar limitations. The structural and algorithmic biases embedded in these systems reflect the constraints of their training and testing environments, often making them ill-suited to adapt to the complexities of real-world operational conditions if they are unforeseen by human designers.

When deployed in unpredictable environments, such systems frequently encounter inputs and situations far removed from their training distributions. Consequently, their engineered emergent behaviours may fail to generalise, as the underlying assumptions and learned patterns no longer hold in these novel contexts. This unpredictability creates significant challenges in ensuring autonomous systems behave consistently and remain trustworthy under all conditions. Addressing these challenges requires robust engineering solutions and a more profound recognition of the parallels between human cognitive limitations and the constraints of artificial systems. Such insights are crucial for designing systems engineering methods to achieve reliable and trustworthy performance in complex, real-world environments. As such we recognise the following challenge:

> **ASE challenge 1:** understanding and predicting “emergence” in a compounded epistemic uncertainty problem involving a harder-to-predict and ascertain system operating in a harder-to-predict and ascertain operational environment is challenging for architects.

C.2.3 ASE problem 2

The risk of missing unforeseeable Black Swan scenarios

There is a risk that during the requirements engineering process, unforeseen, rare, and impactful events—often inaccurately referred to as “edge cases” [12]—may be missed due to their unpredictable nature. These scenarios are particularly challenging to anticipate in the context of autonomous systems operating in complex and evolving environments. Failure to account for such events during system design and requirement engineering poses the risk of system failure in critical situations or behaving unexpectedly, compromising safety, security, and performance. This risk is amplified by the complexity and unpredictability of autonomous systems' intelligent components, leading to potential vulnerabilities and critical failures.

Although standard literature uses the term “edge cases,” we believe this is an inaccurate reference to a well-defined concept in software engineering. The term “edge case” implies that there is a clear boundary that one can easily define. For example, consider a function that only allows numerical input. An edge case might involve a user inputting a 300-digit number. In contrast, a corner case would involve an imaginary number, such as the square root of -1. We clearly understand what constitutes a numerical figure, and the boundary of what is considered a “normal” numerical figure is easily identifiable with no ambiguity. Additionally, we assume that computers are deterministically designed to recognise numerical values.

However, in the context of intelligent components like machine learning-based neural networks, no fixed set of weights definitively determines a numerical value (for example, recognising a digit in an image). The weights in such models continuously change and are entirely dependent on the size and variability of the dataset. So, let me ask you this question:

> *Given that a pictorial visualisation of a human is the base case, what makes an edge or corner case for a pictorial pixel-defined human shape?*

The answer is none. There is no fixed deterministic representation of a human in pixels. There are infinite possible pictorial representations of a human. Hence, we do not use the term “edge cases” to refer to those unforeseeable situations that represent a human for a perception system, for example. Instead, we use the term “unforeseeable Black Swan scenarios or events”. There are two main parts related to the latter concept that we need to clarify:

- **Unforeseeable:** we mean unforeseeable by the ML component relative to what experiences it had been trained on about its operational environment. For example, let's consider an autonomous car designed to avoid hitting birds and land animals. We train the car using 1,000,000,000 labelled images of birds and land animals near roads. However, the machine may face unforeseeable situations, such as encountering a bird on the water's surface. This scenario is absent from the training dataset and, thus, unforeseeable by the machine. It is the rarity of such scenarios that truly test the quality of the model. These are the Black Swan scenarios.

Another rare, unforeseeable scenario could involve a land animal with wings. What if some intelligent land rover car is driving past people wearing wings and bee costumes in the middle of the desert? There are many tourists around the desert areas. Such scenarios become unexpected for the machine if it falls outside the defined expected operational environment. But it is not necessarily an unforeseeable thing for the architect. However, the real risk is whether the architect does not foresee rare scenarios. Hence, we must mitigate this situation by enhancing the architect's predictive capability.

Thus, the term “unforeseeable” does not strictly imply that the scenario is beyond the anticipation of the architect or designer. Instead, it is unpredictable for the machine operating within its expected environmental complexity that the architect defines. For instance, witnessing a horde of gorillas crossing Leicester Square in London would be an unforeseeable event for an autonomous food delivery robot that had never been trained for such a scenario. Still, the design process had only anticipated such a rare scenario that the architect decided not to act on it. This may not be a far-fetched scenario, considering central London is known for its social activities, such as fancy-dress parties. Here, we can appreciate that fancy dresses may not usually be a safety concern in traditional systems. From the robot's perspective, are gorillas considered humans or animals?

- **Black Swan scenarios** are situations that defy conventional expectations or what is deemed a typical operation. They can also be defined as rare but impactful events in a complexity. We use both meanings when we look for Black Swans.

For instance, consider a scenario in which we assume that birds fly away when chased by a dog. A Black Swan scenario could be represented by a dog wearing a fancy dress with butterfly wings fleeing from birds. Another example is a fallen leaf during summer that resembles a helicopter; some trees produce seeds shaped like propellers, which rotate as they fall to disperse the seeds over a wider area. What if we have developed an intelligent autonomous drone designed to intercept other intruding drones or even helicopters trained by micro drones and helicopters? What would such a safety-critical, costly system do in such a scenario?

- Would you have imagined such a scenario?
- How hard do you think it is for you to predict such a scenario?

Black Swan scenarios reflect the complexity of the real world. They are characterised by events whose probability is part of the long-tail distribution assumption of the operational domain rather than a normal distribution. This leads us to believe events are predictable and allows us to overlook the impacts of rare occurrences. In a long-tailed probability distribution profile complexity, infrequent events can have a significant effect, and these are precisely what we refer to as Black Swan scenarios.

Given the above articulation, we define a third challenge that is required to be overcome by our intended systems approach:

> **ASE challenge 2:** establishing confidence that a trained emergent intelligent behaviour will consistently emerge and maintain its designed safety qualities during Black Swan scenarios is a very hard task to demonstrate.

Challenge 3 addresses the difficulty of ensuring that a trained autonomous system's emergent intelligent behaviour (ability to make the right decision in a Black Swan scenario) consistently functions as intended and retains its designed safety attributes across all possible operational scenarios (ideally).

Emergent intelligent behaviour refers to the unpredictable actions, decisions or outcomes arising from the system's interactions with its environment, often influenced by training data and real-world conditions. The challenge emerges from the following factors:

- **Consistency of intelligent emergent behaviour across scenarios**: Autonomous systems operate in environments with infinite variability, including both trained (foreseen) and untrained and out-of-distribution (unforeseen) scenarios. Confidence must be built that the system's behaviour remains reliable and predictable in all these conditions.
- **Preservation of safety constraints during intelligent emergent behaviour**: the system's emergent behaviour must function correctly and adhere to strict safety standards, avoiding unintended outcomes that could compromise reliability or cause harm. We must provide a compelling argument through clever testing to show that a trained model does as expected in Black Swan scenarios.
- **Validation difficulties of how intelligent emergent behaviour will preserve intended constraints**: ensuring safety and consistency requires thorough testing and validation across diverse and unpredictable scenarios. This isn't very easy given the open-ended nature of possible inputs and environmental factors.

This challenge underscores the need for robust predictive thinking frameworks and methodologies to validate and verify that emergent behaviours align with the system's safety and functional goals in a comprehensive and scalable manner.

C.2.4 ASE problem 3

The risk of ineffective management of epistemic uncertainty in the AI training process

To the best of our knowledge, we found that the literature does not discuss the risk of how much ignorance a given finite AI development dataset (like training) has concerning some class system (objects of interest). Ignorance is related to epistemic uncertainty. By epistemic uncertainty, we mean “the difference between the observed distribution of a given finite dataset, for a given operational domain problem, that ideally captures all possible scenarios, to some ideal uniformed, non-biased distribution”. For example, we have a dataset with a, b, and c classes. Each class has 10, 6, or 3 data points. The dataset's distribution (histogram) nature is a biased long-tail distribution. An ideal uniformly distributed dataset may mean the dataset has the same number of 6 different instances for each class (6,6,6). The difference between uniformed and observed distribution makes up an “epistemic uncertainty” of that dataset. This means the dataset has more ignorance in class c than b and is overconfident in class a. Within the captured data, some hidden additional uncertainties exist that the designer might not be aware of. For example, the objects are predominantly captured at the top right corner of the images than anywhere else.

This epistemic uncertainty tells us something about the dataset strategy holistically that impacts the forged intelligence performance in foreseeable and unforeseeable scenarios. Due to the constraints of available resources and the variability inherent in operational environments, acquiring a large, high-quality, and uniformly distributed dataset can prove challenging and costly. Consequently, extensive datasets may be susceptible to hidden biases. It is crucial to comprehend the epistemic uncertainty associated with a given dataset to identify the circumstances in which it may fail to support a reliable intelligent model. To do so, the architect will face two main risks:

1. The risk of hidden epistemic uncertainty related to the operational domain in the dataset for a given classification system (the hidden biases in datasets that capture the real world).
2. The semantic gap between requirements and large datasets hinders the risk of inability to trace requirements down to the dataset level.

C.2.5 Architect’s epistemic uncertainty due to semantic gap

Let’s take an example. Suppose the architect has engineered a dataset. The architect must demonstrate that their large, engineered dataset faithfully captures the required behaviour in given scenarios. This indicates that the claimed epistemic uncertainty is deliberately targeted and reduced. For example, the architect may claim that although the epistemic uncertainty curve of the chosen dataset distribution strategy suggests a non-linear behaviour, tests over unforeseeable validation datasets demonstrated superior performance. The dataset completely covers the requirements. How can the architect show that written requirements are truly captured in those 213,000 images? Mainly including them in a limited safety case. How can the quality assurance team, a regulator, or a court (in case of a catastrophic failure) check all 213,000 images and validate that requirement are truly captured?

There is a significant risk that machine learning datasets for developing ai-based computer vision systems may need to capture or represent the necessary safety and security requirements fully. These datasets are crucial in shaping the behaviour and decision-making capabilities of the AI system. If key safety or security aspects are not included or properly represented during the dataset development, the resulting system may behave unpredictably in real-world scenarios. This can lead to vulnerabilities, including susceptibility to adversarial attacks, biased decision-making, or unsafe operations, compromising the system's reliability, ethical standards, and user safety.

The main challenge of this risk lies in the following premise:

> *It is difficult to close the semantic gap* [13] [14] *between the written description of a requirement (for example, safety) and the infinite possible ways they can appear in a pictorial form.*

In deterministic systems like circuit boards, every component is precisely defined and engineered based on established physical and mechanical properties. Consider the following:

- **Specificity of component characteristics**: for example, based on circuit requirements, resistors can be chosen with exact resistance, tolerance, and power ratings. Capacitors, inductors, and transistors are similarly selected and arranged to achieve the desired behaviours.
- **Trustable predictability**: the arrangement and interaction of these components follow clear principles, allowing predictable and consistent behaviour under known conditions.
- **Trustable implementation of risk mitigation**: engineers can systematically identify and address risks such as electrical interference, overheating, or component failure using rigorous testing and simulation tools. And demonstrate their capture in the design.

The deterministic nature of such systems enables easier trust over their design and performance, making it easier to systematically meet safety and security requirements.

Challenges in machine learning dataset design

In contrast, machine learning systems rely on abstract and often subjective components to be engineered, such as visual representations of target objects of interest (tois), for their design. For example:

- **The problem of specificity: abstract representation of acceptable inputs**. Unlike a physical resistor with well-defined physical properties, a visual representation of a resistor in an image dataset is far less deterministic. Variability in size, orientation, lighting, background, and partial occlusions complicates ensuring consistent identification by an AI system. From a requirement engineering perspective, such a problem poses an intellectual burden in specifying a visual representation of a physical object in a way that can be unambiguous and thoroughly tested. If we select a 10-ohm resistor, we can entirely and unambiguously test such a property using a voltmeter. How can we visually specify a complete representation of a resistor to test a dataset of images to have a resistor in them completely?
- **The problem of trustable implementation of risk mitigation**: **semantic gap between requirements and captured acceptable inputs**. It is particularly challenging to bridge the gap between textual descriptions of requirements (e.g., "identify humans in all scenarios") and their infinite visual manifestations*.*

    - **Example 1**: identifying a human in an image raises questions such as:
        - Does a full-body image qualify as a human representation, or is a hand sufficient?
        - Should the system still classify it as a human if a person is partially visible behind a barrier (e.g., with only a leg or arm showing)?
    - **Example 2**: how should the system handle scenarios where a billboard or other object visually overlaps with a human? If we include it in Multiple images, we technically identify a completely non-human object as part of a human. How will this bias the dataset? What if the billboard has unethical gratification on it? Or images of an animal or another car? Now, we have introduced less-priority objects in terms of safety into the image of a human.
    - Why would such a scenario be important? An autonomous car must accurately differentiate between a human and a billboard. If a billboard visually overlaps a partially visible human, the system must not ignore the human presence, as doing so could result in a dangerous collision.

ASE challenge 3: it is very hard to predict, trace, and risk-manage the share size of infinite acceptable input scenarios associated with an open, complicated environment interacting with an autonomous system and vice versa.
What do we mean by "***infinite acceptable inputs***"?

Let's take a simple light switch as an example; the acceptable inputs of a light switch are an electrical current and a mechanical input that is pressed either on or off. A car engine is more complex than a light switch; it accepts a wider range of inputs. Nonetheless, the set of all possible acceptable inputs is specifiable and finite in both cases and can be measured precisely. However, when considering a perception system run over a neural network, which is tasked with detecting a flying drone, for example, there are infinite ways that a flying drone may visually appear as input. Each possible way is classed as an acceptable input. Therefore, systems engineers must specify infinite possible inputs in a traceable manner that cannot be precisely measured.

Therefore, in some high-performance cars, we can develop a far more trustable assurance case for a highly complicated car engine than we could for a neural network with no moving parts. This says something about our understanding of what simplicity and complexity mean and the role of the problem-solver's ability to predict.

References

[1] INCOSE, SYSTEMS ENGINEERING HANDBOOK, INCOSE, ISBN: 978-1-119-81431-3, 2023.

[2] INCOSE, IEEE Systems Council, “Brief History of Systems Engineering,” SEbok , 20 May 2022. [Online]. Available: https://www.sebokwiki.org/wiki/Brief_History_of_Systems_Engineering. [Accessed 15 Sep 2022].

[3] P. Checkland, Soft systems methodology: a 30 year retrospective and systems thinking,, Chichester: Wiley, 1999.

[4] H. Sillitto, D. Dori, R. M. Griego, S. Jackson, D. Krob, P. Godfrey, E. Arnold, J. Martin and D. McKinney, “Defining “System”: a Comprehensive Approach,” in INCOSE International Symposium, 2017.

[5] D. Rousseau, “Advances in the Prospects for Realizing a Scientific General Theory Underpinning Systems Engineering,” in 2019 International Symposium on Systems Engineering (ISSE), Edinburgh, UK, 2019.

[6] H. Al-Shareefy and S. Wright, “Applied systems science to holistic quality assessment metrics for formal methods-based models,” in 68th Meeting of the International Society for the Systems Sciences, Washington DC, USA, 2024.

[7] A. Tversky and D. Kahneman, “Judgment under Uncertainty: Heuristics and Biases,” Science, vol. 185, no. 4157, pp. 1124-1131, 1974.

[8] N. G. Leveson, Engineering a Safer World, London, UK: MIT Press, 2011.

[9] PQRI, “Hazard & Operability Analysis (HAZOP), Risk Management Training Guides,” PQRI.

[10] P. Kalia, M. Menzel, K. Grello and A. Walker, “Integrated Systems Engineering, Safety, Reliability and Risk Management – Minimizing Black Swan Events,” in 2024 Annual Reliability and Maintainability Symposium (RAMS), Albuquerque, NM, USA, 2019.

[11] A. Tversky and D. Kahneman, “Judgment under Uncertainty: Heuristics and Biases: Biases in judgments reveal some heuristics of thinking under uncertainty.,” Science, pp. 1124-1131, 1974.

[12] Carnegie Mellon University, “Phil Koopman,” Carnegie Mellon University, [Online]. Available: https://users.ece.cmu.edu/~koopman/. [Accessed 24 06 2022].

[13] S. Burton and B. Herd, “Addressing uncertainty in the safety assurance of machine-learning,” Front. Comput. Sci., vol. 5, 2023.

[14] P. W. M. Koopman, “Challenges in Autonomous Vehicle Testing and Validation,” in 2016 SAE World Congress, 2016-01-0128 / 16AE-0265, Detroit, 2016.