Table of Contents

Appendix F Detailed Summary of AIC Systems Approach Processes

In this section, we will summarise our process with limited examples to give a general appreciation of what the design approach entails. Then, we will perform each case study in Appendices H, I. The idea behind our design approach is to imagine that we are re-organising or engineering the entire problem domain from the first principles of systems, with the autonomous system being introduced into the mix. We assume that the problem domain is a “Confusing Complex”, and now we need to re-design it with an autonomous system solution within it and turn it into a close as possible to an “Ideal System”. Therefore, we need to be very careful not to end up causing unintended consequences that could lead to further unwanted confusing complexity and end up in a situation where we cause broader problems than solving. After finishing the design process, we will create a partial safety case using SACE and AMLAS patterns in Appendix L.

The design artefacts in regular systems safety cases include arguments around traditional safety analysis techniques such as FTA, FMEA, and HAZOP for the whole system of interest. In our research, we will focus only on the safety assurance of the intelligent component and ignore the traditional deterministic subsystems for future research.

F.1 Predictive Complicatedness Resolution Matrices

The clarity of complicatedness is relative to the predictive architect's knowledge and thinking framework. To increase the effectiveness of any complexity field resolution approach, the approach must help the architect resolve confusion about any complicatedness and lead them to relative clarity. One way to do so is to enable the architect to ask themselves general predictive questions in any observed complexity field. However, we need a structure that captures the map of complexity fields, so we use predictive thinking matrices.

> **Why are they "predictive thinking" tools?** Because they prompt the architect to make a prediction about some complexity of events based on a first principles approach (considering all possible interactions) and general systems patterns or rules, to predict how and what interactions will occur in a complexity field.

Architect's Predictive Complicatedness Resolution Matrix (ArcMatrix) is a predictive thinking tool that serves as a visual and structured representation of complicated relationships (of some complexity field) in a matrix-like manner, enabling architects to resolve the complicatedness of any observed confusing complexity by understanding and dissecting complicated interactions and their consequences holistically. The matrix ensures that the complex's design considers all possible observable interaction scenarios among complexes. It is an analytical tool designed to capture a macroscopic view of complicated relationships accurately. While solving the Predictive Matrix, the architect must make assertions and predictions about the observed Confusing Complex, thus aiding with predictive thinking.

AIC systems philosophy touches on the notion of "making the invisible visible". This means that AIC theory focuses on making justified objective assumptions about the categories of knowledge that are invisible in any observed complexity, visible.

And what critical invisible things keep Compounded Uncertainty Problems architects awake at night? The answer is:

Black Swan Scenarios

We interpret this notion in our adapted AIC systems approach as follows: one goal of predictive thinking processes is to objectively make the invisible complicatedness among Black Swan Scenarios of a complexity, visible, explainable and justified. ArcMatrix is one process that transitions ArcPM from a situation of unknowing the invisible dynamics of a complexity field to having justifiable knowledge of what is going on among the coexisting elements. To such an end, we devise two levels of rigour in solving the confusion of complicatedness: ***superficial*** and ***deep***. We have two types of ArcMatrix: Simplified Matrix (Actions Matrix) and Deep AIC Complicatedness Matrix (AIC-Matrix).

F.1.1 The Deep ArcMatrix (Deep AIC-Matrix)[1]

Knowing that an intelligent drone is trespassing into a train track zone is not enough information to make a sound engineering judgement (prediction) on how to deal with it, because the nature of intelligence opens up a can of possible reasons why it is doing so. One of the most critical risks of intelligence is “intention”. Suppose we deal with regular non-learning complexes, such as track zone electric power lines. In that case, they have one goal: to remain securely attached to the infrastructure to ensure the delivery of electrical power. Learning complexes, such as intelligent drones, can adapt their goal settings depending on their situation. Therefore, knowing only that there are access train tracks is not enough and more assumptions need to be clarified. Here is where the AIC-Matrix come in to solve.

Key Components of the Deep AIC-Matrix:

1. **Rows (Source Complexes)**:
    - Represent the entities initiating interactions or actions in the complex. These may include intelligent agents (e.g., drones) or non-intelligent systems (e.g., power lines).
2. **Columns (Sink Complexes)**:
    - Represent the entities receiving the effects of the interactions. Sink complexes may include operational zones, physical infrastructure, or other agents in the environment.
3. **Cell Details (Interaction Emergence Decomposition)**: Each cell in the matrix captures the interaction between a specific source and sink complex, decomposed into the following elements:
    - **i. Supra Source**: Describes the encompassing supra complex (larger complex) from which the source complex emerges or operates. This provides a complexic view of the source's operational context.
    - **ii. Supra Source PrimeP**: Details the encompassing **PrimeP** (primary purpose) of the source complex within its supra source. This clarifies the higher-order intent driving the source's behaviour.
    - **iii. Source Goal**: Identifies whether the source complex has a defined goal related to the sink complex. If no goal exists, it is explicitly noted as "no-goal."
    - **iv. Source Goal Type**: Categorizes the goal as either an **AIC-type goal** (appreciative, influence, or control) or "unknown" if the goal type cannot be classified.

[1] The tool used in section 6.6.2.

- **v. Source Action**: Specifies whether the source complex takes an action toward the sink complex. If no action exists, it is noted as "no-action."
- **vi. Source Action Type**: Categorizes the action as an **AIC-type action** (appreciative, influence, or control) or "unknown" if the action type is unclear.
- **vii. Source Action Effect Type**: Evaluate the effect of the action on the sink complex:
  - **Supportive**: The action positively contributes to the sink complex's goals or operations.
  - **Obstructive**: The action creates challenges or hinders the sink complex.
  - **Neutral**: The action has no significant effect.

Below is a general representation of Deep AIC-Matrix:

Table F.1 General format of Deep AIC-Matrix

	Sink complex A	**Sink complex B**
Source complex A		**Supra Source:** encompassing Supra Complex. **PrimeP:** encompassing Supra Complex PrimeP **Goal:** goal, or no-goal **Goal type:** AIC or unknown. **Action:** action or no-action. **Action type:** AIC or unknown. **Effect:** Obstructive, Supportive, Neutral.
Source complex B	**Supra Source:** encompassing Supra Complex. **PrimeP:** encompassing Supra Complex PrimeP **Goal:** goal, or no-goal **Goal type:** AIC or unknown. **Action:** action or no-action. **Action type:** AIC or unknown. **Effect:** Obstructive, Supportive, Neutral.	

Table F.1 introduces the **Deep AIC ArcMatrix (AIC-Matrix)**, a framework designed to decompose and analyse the complicated interactions among intelligent and non-intelligent systems within an operational context. It is particularly relevant for learning complexes scenarios, where their adaptability and potential intentions must be understood to ensure effective engineering judgments and complex safety. The AIC-Matrix provides a structured mechanism for evaluating the relationships between **source complexes** (entities initiating actions or goals) and **sink complexes** (entities receiving the effects of these actions or goals). Each matrix cell details the nature of the interaction between a source and a sink by decomposing it into well-defined aspects. For example, the following is an example application of the AIC-Matrix:

Table F.2 Eagle Robot case study example of applying a Deep AIC-Matrix to describe the relationship between train track zone and anonymous drone

	anonymous drone	**train track zone**
anonymous drone		**Supra Source:** Adversarial Scheme. **PrimeP:** Disrupt Train Network operations. **Goal:** Strike passing train. **Goal type:** Control. **Action:** Intrude. **Action type:** Control. **Effect:** Obstructive.
train track zone	**Supra Source:** Train Network. **PrimeP:** Safely transport people and goods. **Goal:** manage drone access. **Goal type:** Appreciation. **Action:** no-action. **Action type:** Appreciation[2]. **Effect:** Supportive.	

Table F.2 illustrates an application of the Deep AIC-Matrix to analyse the interaction between an anonymous drone and the train track zone within the Eagle Robot case study. The matrix captures

[2] A no-action is an appreciative type behaviour, since no-action entails no influence or control, and place the system in an appreciative situation.

the relationship between these two entities, with each cell detailing the interaction's goals, actions, and effects. The anonymous drone is identified as part of an Adversarial Scheme with a Prime Purpose (PrimeP) of disrupting train network operations. Its goal is to strike a passing train, categorised as a Control-type goal, executed through an intrusion action of the control type. This action is obstructive, signifying a clear threat to the train network. In contrast, the train track zone is part of the Train Network Supra Source, with a PrimeP to ensure the safe transportation of people and goods. Its goal is to manage drone access, which is classified as an appreciative-type goal, as it takes no direct action and relies instead on appreciative oversight. This lack of action supports the drone's obstructive action, which does not align with the train track zone’s overarching safety and operational continuity purpose. This example demonstrates the utility of the AIC-Matrix in breaking down and understanding complicated interactions between intelligent agents and non-intelligent systems.

The case study problem situationment does not explicitly define the drone's purpose in the given example. Our assumption that the drone's purpose is adversarial is based on its obstructive effect on the safety of the train track zone for passing trains. However, it's essential to justify such assumptions in practice. Constructing the AIC-Matrix requires a Chain-of-Thought process to process information from the Actions Matrix to arrive at the answers in the AIC-Matrix. Therefore, the process of resolving the AIC-Matrix requires us to predict the adversarial nature of the drone. Additionally, the zone does not influence or control drone action; it only allows it to happen due to the absence of action. Considering the PrimeP, we can deduce a goal to manage drone access, which is an appreciative goal due to a lack of capability. If the current goal is to appreciate, then the inaction that results in an inability to influence or control the drone's behaviour is also of an appreciative nature. Consequently, the output effect of the train track zone's inaction behaviour supports autonomous drone intrusion.

Note how we elaborated a detailed and comprehensive scenario out of a single action word, “permit access.” If we read the scenario, knowing AIC intuition, we can validate the reasoning behind the assumption. While resolving the complexity using AIC-Matrix, we made predictions about the observed situation. Hence, AIC-Matrix adds an artefact demonstrating sound articulation of a problem and explaining decisions made.

F.1.2 Actions Matrix[3]

When facing a Confusingly Complicated situation, the ArcPM may find it helpful to construct an initial, simplistic appreciation of the observation. Such appreciation allows the architect to consider a bird's-eye view (holistic) picture of what is being observed, abstracting out the details.

[3] The tool used in section 6.2.2 The Eagle Robot case study.

The choice of complexes to be considered as part of the initial assessment is not subjective but objective, due to the work done before their construction. For example, we introduce the superficial matrix in Predictive Thinking Pipeline 2 in the first stage, after objectively identifying the interacting complex in Predictive Thinking Pipeline 1. The matrix is constructed by mapping all elements of the concerning complexes to each other, such that each complex is mapped to all complexes in the observed whole. The main components of the Actions Matrix construct are:

a Source complexes (represented in matrix rows).

b Sink complexes (represented in matrix columns).

c Source Actions in every cell.

The following table represents a generic format of Actions Matrix:

Table F.3 General Format of Actions Matrix

	Sink complex A	**Sink complex B**
Source complex A		Source complex A Action
Source complex B	Source complex B Action	

Table F.3 presents a generic format of the Actions Matrix, which is a quick tool for resolving the predictability of complexity fields by analysing the specific actions taken by source complexes in relation to sink complexes. The rows represent the source complexes initiating the actions, while the columns represent the sink complexes that are the targets or recipients of these actions. Each cell in the matrix captures the specific action undertaken by the source complex towards the sink complex.

For example, suppose the following problem scenario: An anonymous Drone intruded into a bounded train tracks zone. To construct an Actions Matrix, first, we define the observed complexes of concern set: {anonymous drone, train track zone}. An action matrix can be constructed into the following:

Table F.4 Example Actions Matrix

	anonymous drone	**train track zone**
anonymous drone		Intrude
train track zone	Permit access	

Table F.4 illustrates an Actions Matrix applied to a scenario involving an anonymous drone and a bounded train track zone. The scenario begins with identifying the observed complexes of concern: {anonymous drone, train track zone}. The Actions Matrix is then constructed to capture the specific interactions between these complexes. As a source complex, the anonymous drone executes the action of intrusion into the train track zone, reflecting a direct and deliberate engagement with the complex. Conversely, the train track zone, as a source complex, acts as permitting access, indicating a passive response that allows the drone to continue its intrusion. This matrix effectively captures the dynamics of the scenario by isolating the actions of each complex and demonstrating their interaction.

Note that the initial description of the problem did not explicitly expose what the train track zone is doing to the intruding drone. Here, we had to think harder to predict or assert what action track zone is performing towards the drone. Such aspects remain invisible from the explicit accurate description as a tacit assumption. The Actions Matrix forced us to consider carefully what kind of action the train track zone is doing to the drone. Therefore, the Actions Matrix tell us that our initial perception of the problem was incomplete. However, there is more invisible knowledge about "drone intrudes into the zone, and zone permit access of drone", which we need to make explicit and visible. Here is where the process of constructing an AIC-Matrix comes in handy.

The following is an example taken from our AVOIDDS case study, which shows how AIC actions matrix looks like:

Table **Error! No text of specified style in document.**.5 Example AIC Actions Matrix for AVOIDDS problem domain. Red is unresolved influence, Blue is control, Green is appreciation, Black is resolved influence.

									
		+capture	+capture	+assure	+assure	+capture	+predict	+capture	+validate
	-visually complicate		visually complicate	visually complicate	-operationally complicate	+control	-reduce reliability	visually complicate	-complicate
	-visually complicate	operationally complicate		visually complicate	operationally complicate	operationally complicate	-reduce reliability	-operationally complicate	visually complicate
	+predict	+predict	predict		+assure	+predict	+optimise	-predict	+validate
	+inform	+monitor	monitor	+inform		+inform	+monitor	+control	+inform
	-visually complicate	+expose	visually complicate	-visually complicate	-visually complicate		-reduce reliability	visually complicate	visually complicate
	+predict	compute visual	+compute visual	+assure	+assure	+compute visual		+compute visual	+compute

		complexity	complexity			complexity		complexity	
	-visually complicate	visually complicate	-visually complicate	-visually complicate	-operationally complicate	visually complicate	reduce reliability		visually complicate
	+qualify	+expose	capture	+assure	+assure	+capture	+predict	+capture	

Table 5.24 captures the AIC complexity field for the AVOIDDS problem domain, where all interactions are accounted for. Note, we also included the architect as part of the problem.

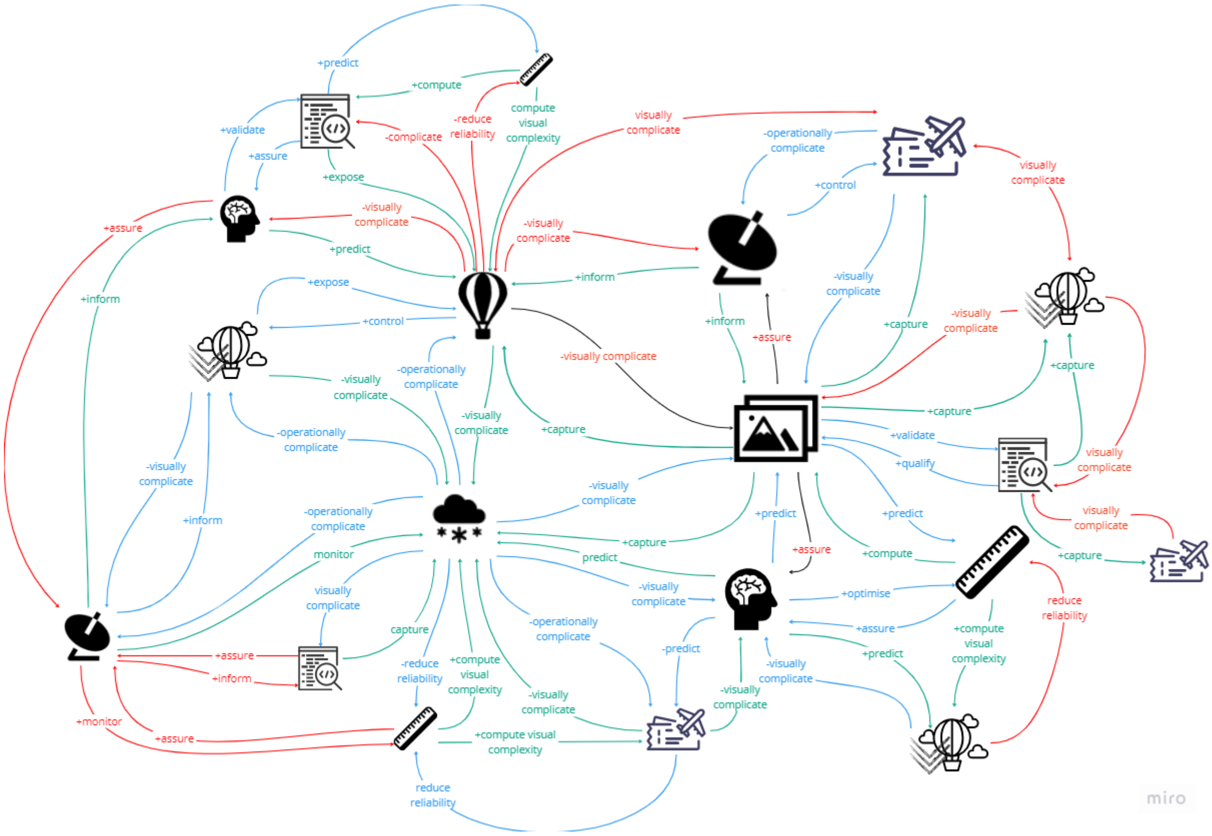


Figure **Error! No text of specified style in document.**.1 AIC complexity field of AVOIDDS problem

The icons are defined in the following table, which captures the main sources of complexity:

Table **Error! No text of specified style in document.**.6 AVOIDDS problem complexes icons definition

Situation	**Icon**
other_flying_objects	

weather_conditions_variety_definition	
accurate_architect_predictive_mental_model_ArcPM	
informed_air_traffic_controller	
avp_development_datasets (avp stands for aircraft avoidance perception system)	
reliable_correct_target_object_recognition_confidence_scores	
critical_data_accuracy	
visual_traffic_density_and_patterns	
other_flying_objects _visual_complexity_parameters	

Overall, the AIC-Matrix framework provides a structured approach to resolving complicated interactions in a long-tailed complexity field (abnormal complexity). It enables the architect to predict how desired systems should behave, predict Black Swan scenarios and enables more resilient assumptions in dynamic and uncertain environments.

F.2 AIC Emergent Relationship Factorisation Process

AIC emergence factorisation SECoT is a complicated interaction analysis, rethinking and development process. Applying AIC factorisation within a Chain-of-Thought framework provides a systematic process for dissecting the emergence of complex interactions into the pattern of causal factors. This approach not only assists with unpacking current complicated capabilities but also aids in predicting and preparing for potential future challenges by exploring various perspective shifts. Table F.5 outlines a structured SECoT approach to facilitate a deeper understanding of complicated interactions within a specified environment, mainly focusing on the dynamics of AIC (Appreciation, Influence, Control) factorisation of complicated relationships. The process involves five steps: identifying the source and sink complexes, determining the action undertaken by the source on the sink, categorising the action as appreciative, influencing, or controlling, defining the likely purpose behind the source's behaviour as appreciation, influence, or control, and assessing the effect of the source's action on the sink's primary purpose, whether supportive, obstructive, or neutral.

Table F.7 AIC emergence factorisation SECoT process

SECoT Title	AIC factorisation of complicated relationships in an environment
SECoT Input	A description of some phenomenon (an interaction or a scenario) For example, Eagle Drone synchs its internal GPS module with the Global Navigation Satellite Complex.
General Systems Rules	**AIC complicatedness:** A complex's relationship is formed when there is a source and a sink; the source can implement an effective action upon the sink. All complexes perform actions with a purpose that may be known or yet to be known. There are three types of effects: Any action taken by a source complex can impact a sink complex's chances of manifesting its primary purpose in a given scenario or environment, such as a supportive, obstructive, or neutral complex.
Predictive **Predictive Thinking Process**	**Step 1) Predictive Question:** What is the interaction's source and sink complexes? **Guiding Prompt:** Review the observed situation and identify a source and sink complex. For example, **Source:** Eagle Drone. **Sink:** GPS module.
	Step 2) Predictive Question: Given the source complex, what is the larger Supra Source and PrimeP? **Guiding Prompt:** Define the Supra Source complex of which the source is part. For example, **Supra Source:** Policing Force. **PrimeP:** Ensure the safety and security of the Train Network.
	Step 3) Predictive Question: What action is taken by the source upon the sink? **Guiding Prompt:** Identify an action in the format of a very in the present tense that is being undertaken by the observed source. For example, **Action:** Synchs
	Step 4) Predictive Question: What is the AIC type of the action? **Guiding Prompt:** Specify the type of actions (Appreciative, Influencing or Controlling) Example, **Action type:** Controlling

	Step 5) Predictive Question: What is the most likely source's goal behind the source behaviour and its AIC type? **Guiding Prompt:** Given the context of the observed interactions, define the most likely source's goal and its AIC type. For example, **Goal:** Geolocate its position within the track zone area. **Goal Type:** Control.
	Step 6) Predictive Question: How does the source complex's action affect the sink's behaviour? **Guiding Prompt:** Define the most likely effect the identified action has on the sink's chances of manifesting its primary purpose. Example, **Effect type:** Neutral.
Output prediction	Capture the answers to the above thinking steps in the list: **Source:** An affecting complex whose behaviour is the source of interaction. **Sink:** An affected complex by the behaviour of the source. **Supra Source:** A larger encompassing complex where the source is part of. **PrimeP:** Source complex primary purpose. **Action:** Action being taken by the source. **Action type:** Appreciative, Influencing or Controlling. **Goal:** Source's intended end situation that is expected to happen. **Goal type:** Appreciation, Influence or Control. **Effect type:** Source action effect on sink.

F.3 SECoT_1: An AIC-based Compounded Epistemic Uncertainty Problem Articulation[4]

The study in [1] explores how prompting techniques enhance the ability of large language models to perform complex reasoning tasks. In our research, we realised that when we use CoT techniques with General Complex Theories, we can systematically work through the complexities of open operational environments. This systematic approach allowed us to predict and justify our assertion about potential outcomes based on interactions and changes between the complex and its operational environment. Every step in our CoT process consists of a predictive question that triggers the thinking operation and a predictive guiding prompt that assists with solving the question. Building on this foundation, our research introduces the adoption of the AIC framework

[4] See sections H.4, I.4 for full implementation.

as an integral component of the Chain-of-Thought (CoT) process. The AIC-based Chain-of-Thought (AIC-based CoT) process applies the AIC systems approach to improve confidence in predictive reasoning.

Therefore, the solution of every step is an assertion claim: "The architect asserts that ...". The steps in the process are iterative, which means the architect seeks clarification and help to answer the question and solve the prompt while refining their prior assumptions made in the earlier step. Architects may carry on in each step until no new information can be added. We will present the process by defining the general steps to be taken and explain how each step was applied in the case study. A full version of the analysis can be found in [2].

F.3.1 Predictive Thinking Pipeline 1: Appreciate the Complexes of the Problem Complexity Field

In this stage of the thinking process, the architect focuses on applying the first principle by identifying the primary purpose through the following goals:

- Recognising the systems of concern related to the problem.
- Identifying the Supra Complexes that include these systems.
- Defining the primary purpose of these Supra Complexes.

Additionally, we must clarify the notion of Confusing Complexes. The problem often stems from unclear complexes' purposes or conflicting objectives that hinder the architect's effort to realise their intent. The main goal in our practice is to adjust the complexity dynamics to empower a complex of concerns to manage the situation effectively. In this process, we build a very carefully crafted deep description of what is the nature of the complexity field we are trying to handle. We need to predict the future of the complexity field after deploying our solution but to do so; we need a deeper understanding of the present by understanding the purposes and general rules of systems.

a) Step 1.1) Identify a list of unsafe or confusing behaviours:

This step aims to identify and articulate unsafe or confusing behaviours within a specified problem domain, particularly those leading to undesirable emergent outcomes. The process begins with defining "unsafe behaviour" as a subset of "confusing behaviour," where confusing behaviour contradicts expected or intended system outcomes. The architect uses Predictive Thinking Processes to evaluate scenarios where these behaviours disrupt the system's purpose or harmony. A guiding prompt focuses on uncovering aspects of the problem that break systemic harmony, such as contradictory actions within the operational domain. For example, a drone hovering near a restricted train track zone without demonstrating overtly hostile behaviour represents a "confusing" situation. Although not immediately dangerous, such behaviour undermines security protocols, normalises intrusions, and risks desensitisation to real threats. The step concludes successfully when problematic scenarios and their impacts are documented, forming a list of unsafe and confusing behaviours to inform further system safety design.

Table F.8 SECoT for Predicting Unsafe Behaviours

SECoT Title	Unsafe problematic behaviours list identification
Input	Problem brief
General Systems Rules	**General rule A**: Unsafe behaviour is a type of Confusing Behaviour. Confusing behaviour of some situation A, in any complexity, is a behaviour that seems contradictory to what is meant to be manifested. **General rule B:** Confusing behaviours lead to undesirable emergent outcomes about some elements (of the observed complexity) with respect to some element B, whereby situation A falls within element B's sphere of concern for others.
Predictive Thinking Process	**Predictive Question 1.1.1:** What unsafe behaviours within the problem domain's Confusing Complex disrupt an expected manifestation of some purpose, potentially leading to undesired consequences, which fall into our sphere of concern? **Guiding Prompt 1.1.2:** Identify an aspect of the problem of a single informative concern which you deem to be unsafe or confusing as it does not follow an intended purpose or seems to break some desired harmony in a system. We defined the term "confusing" to include scenarios where the behaviour may not be "unsafe", for example: *A drone is observed hovering near a restricted train track zone for an extended period. Still, it does not approach any critical infrastructure or demonstrate hostile behaviour, such as approaching a moving train or interfering with surveillance cameras. Instead, it stays at a fixed altitude, maintaining a distance from the tracks and infrastructure*

	while intermittently adjusting its position within a defined perimeter. In other words, an informative concern is a problematic situation in which some system of interest is undesirably behaving in a way that contradicts the expected PrimeP. This step is considered competitive when a problematic aspect, including its causal impact, is identified. **Step completion criteria 1.1.3:** The step is regarded as complete when a problem situation is described, including undesirable impact.
Output Prediction	**Architect assertion 1.1.4:** List all problematic and unsafe situations

To further outline the meaning of the SECoT, we will use the following GSN format:

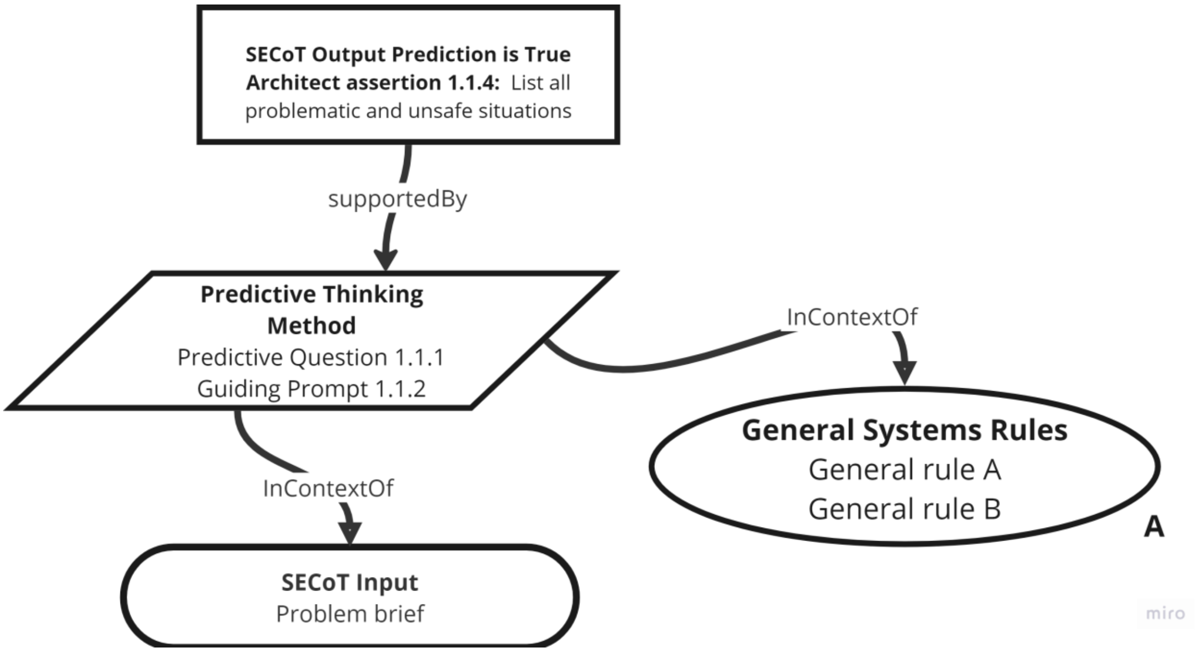


Figure F.6b GSN formatted SECoT

It is possible to communicate the SECoT formally using the GSN format. Figure F.6b provides an example of how to describe a SECoT or any predictive thinking process. A predictive thinking process requires context, which includes the input and assumptions; in this case, these would be the general systems rules, and the asserted output is architectural prediction or assertion.

b) Step 1.2) Generate a descriptive image that visualises the unsafe behaviour

This step focuses on creating a visual representation of the unsafe or confusing behaviour identified in Step 1.1. The goal is to synthesise a graphical depiction that captures the problematic situation's essence, illustrating the conflicting interactions and emergent outcomes contributing to its complexity. Building upon the architect's assertions from Step 1.1, this process aligns with the principles of General Systems Theory, wherein unsafe behaviour is conceptualised as a type of "confusing behaviour" that disrupts intended systemic harmony. The guiding process addresses a predictive question: *How does the confusing behaviour visually manifest within the problem domain?*

The architect is encouraged to employ abstract visualizations, using text-to-image generation tools (e.g., DALL-E), to iteratively refine a depiction that encapsulates the observed complexities. This visualization must integrate elements demonstrating how the behaviours intersect within the architect's and stakeholders' spheres of concern. For instance, an image might depict a drone hovering near restricted train tracks, showing its proximity to critical infrastructure, surveillance cameras, or human activity, highlighting the unsafe and ambiguous aspects of the behaviour. The process is complete when a single, sufficiently detailed visual effectively communicates the interrelations of key elements and their emergent impacts. The resulting image serves as a foundational artefact for deeper system analysis and validation in subsequent steps, reinforcing the architect's comprehension of the unsafe situation within the operational complexity field.

Table F.9 SECoT for problematic situation visualisation

SECoT Title	Problematic situation visualisation
Input	Architect assertion 1.1.4
General Systems Rules	**General rule A**: Unsafe behaviour is a type of Confusing Behaviour. Confusing behaviour of some situation A, in any complexity, is a behaviour that seems contradictory to what is meant to be manifested. **General rule B:** Confusing behaviours lead to undesirable emergent outcomes about some elements (of the observed complexity) with respect to some element B, whereby situation A falls within element B's sphere of concern for others.
Predictive Predictive Thinking Process	**Predictive question 1.2.1:** How does the confusing behaviour appear visually? **Guiding prompt 1.2.2:** Graphically visualise how you perceive the problem, confusing the whole scenario encompassing the problem aspect. You may use text-to-image generation tools such as Dall-E to generate an abstract, simplistic sketch representing how you imagine the situation. You may experiment with different prompts until you find an appropriate (detailed yet realistic) representation of the problem as you picture it.

	Step completion criteria 1.2.3: The step is considered complete when a single appropriate visual representation visualises how parties to the problem are within each other's sphere of concern and how the model depicts a situation within the architect's sphere of concern.
Output prediction	**Architect assertion 1.2.4:** produce an abstract depiction of the unsafe situation.

c) **Step 1.3) Define the complex of the complexity field:**

This step systematically identifies and defines the "complex of complexes" constituting the problematic situation, building on prior architectural assertions and visual representations. Following the principles of General Systems Theory, this step operationalises General Rule C, which posits that any complexity field comprises coexisting elements, whether or not they actively interact. These elements, collectively called a "complex of complexes," form the structural foundation of the observed problem domain.

The process begins with analysing the architect's assertions from Step 1.1 and the visual depiction generated in Step 1.2. Guided by the predictive question, *"What is the complex of coexisting elements involved in the problematic situation?" the* architect identifies the key elements that coexist and contribute to the emergent complexities. The guiding prompt directs the architect to use the visual representation as a cognitive tool to infer the constituent elements of the scenario. This involves deliberately examining these elements' roles, relationships, and boundaries within the observed problem domain.

The step is complete when a comprehensive list of these coexisting elements is defined and can be mapped to features in the visualised model. For example, in the case of a drone hovering near restricted train tracks, the defined complex may include elements such as the drone, the train tracks, surveillance cameras, the train itself, and nearby human actors. This step ensures that the architect's understanding of the complexity field is explicitly articulated and structured, laying the groundwork for subsequent analyses. The output of this step is codified as an assertion, delineates complex concerns, and establishes a detailed baseline for addressing systemic interactions and emergent behaviours in the problem domain.

Table F.10 SECoT for complex of complexes definition

SECoT Title	The complex of complexes definition
Input	Architect assertion 1.1.4 Architect assertion 1.2.4
General Systems Rules	**General rule C:** Complexity is a field containing an organisational experience of a phenomenon concerning a general problem-solver (such as a Predictive architect). It involves an operational environment of coexisting complexes and the complicated nature (or types) of their relationships and interactions (epistemic uncertainty) for a Predictive architect to predict their past, present, and future situations (managing aleatoric uncertainty or randomness).
Predictive Thinking Process	**Predictive question 1.3.1:** What is the Complex of coexisting elements involved in the problematic situation, as observed in the model? **Guiding prompt 1.3.2:** Guided by the visual representation, infer a complex of complexes you imagined to be part of your perceived scenario. **Step completion criteria 1.3.3:** The step is considered complete when a list of elements is defined and can be identified in the visualisation model.
Output prediction	**Architect assertion 1.3.4:** The architect asserts that: List of complexes of concerns involved in the situation

d) Step 1.4) Define the Supra Complexes and their PrimePs

This step aims to establish the hierarchical context of the previously defined complex by identifying its encompassing Supra Complexes. A Supra Complex, as guided by the AIC Systems Approach, represents a higher-order system or collection of complexes that includes the defined complex as a constituent part. This process is rooted in the General Systems Rules, which posit that every complex exists within a broader supra-structure, termed Supra Source (encompassing source complexes) or Supra Sink (encompassing sink complexes), depending on its role in the interaction dynamics.

The process begins by revisiting the architect's previous assertions regarding the problematic complex and its visual and structural elements. The predictive question, *What are the interacting Supra Complexes of concern of which the coexisting elements are parts?*, guides the architect to conceptualise the broader systems that integrate and govern the identified complex. Using the guiding prompt, the architect examines the common purposes and interactions of the defined complexes and extrapolates their roles within larger, cohesive systems, denoted in capitalised format to signify their supra-context (e.g., Transport Network, Security Framework). The step is considered complete when a comprehensive list of Supra Complexes is defined, each encompassing the visible elements of the problem domain.

As for establishing the System's Primary Purpose (PrimeP) for each Supra Complex identified in the previous steps, the General Systems Rules, Rule E and Rule F establish that the PrimeP represents a Supra Complex's consistent and overarching purpose, ensuring alignment of the architect's focus when predicting behaviours across all possible scenarios. An Ideal System remains consistently purposeful(meaning its functionality and influence do not become obsolete due to changes in the complexity field), with its PrimeP clear and comprehensible to any observer.

The predictive question, *What is the PrimeP for every Supra Complex such that it is expected to manifest in all possible scenarios?*, directs the architect to deduce a unifying purpose for each Supra Complex that encapsulates its intended functionality and objectives. Using the guiding prompt, the architect carefully considers each Supra Complex's operational context and desired outcomes to infer a purpose that aligns with broader system goals while shaping subsequent design priorities. The step is complete when a clear and actionable PrimeP is defined for each Supra Complex, ensuring all design and engineering efforts align with these foundational priorities.

Table F.11 SECoT for Supra Complexes definition

SECoT Title	Supra Complexes and their PrimePs definition
Input	Architect assertion 1.3.4
General Systems Rules	**General rule D:** Supra Complex, Supra Source and Supra Sink, AIC Systems Approach. A **Supra Complex** is a relatively larger collection of complexes where a complex of interest (of a predictive observer) is part of. **General rule E: Requisite consistent purpose:** An Ideal System is forever consistently purposeful, which means having a clear purpose with respect to the architecting observer that is clear to all possible observers and in all possible scenarios. **General rule F:** System Primary Purpose (PrimeP), AIC Systems Approach

<table>
<tr>
<td>Predictive Thinking Process</td>
<td>
Predictive question 1.4.1a: What are the observed interacting Complexes, the possible holistic PrimePs they serve, and their Primary capability (or functions) that define their operational situation?
Predictive question 1.4.1b: What are the interacting Supra Complexes of concern of which the coexisting elements are parts?
Predictive question 1.4.2: What is the PrimeP for every Supra Complex such that it is expected to manifest in all possible scenarios?
Guiding prompt 1.4.3: to help with answering 1.4.1a, used the following table:
<table>
<tr><th>Observed System</th><th>Primary Purpose (PrimeP)</th><th>Primary Capability (PrimeC)</th></tr>
<tr><td>Complex</td><td>Possible holistic PrimeP to serve</td><td>The primary function that delivers the PrimeP is written in the format of {adjective_noun}</td></tr>
</table>
As for 1.4.1b, consider the identified systems as parts of Supra Complexes. Then, define the associated Supra Complexes, encompassing systems with a common purpose. Supra Complexes are written in a capitalised format.
Guiding prompt 1.4.4: Infer the PrimeP for every Supra Complex. The choice of the PrimeP will guide the rest of the design on what priorities each Supra Complex intends to achieve. Getting the priorities focused only on specific aspects or generally on broader aspects dictates the design scope. For example, suppose we have a Road Transportation as the Supra Complex. Suppose we choose a PrimeP to mobilise people across a single specific strip of road. In that case, we will constrain the design decisions to be influenced by the limited, finite stretch of the operational domain.
Step completion criteria 1.4.5: The step is considered complete when a list of Supra Complexes encompasses all the observed visible elements. Also, a PrimeP is defined for each Supra Complex.
</td>
</tr>
<tr>
<td>Output Prediction</td>
<td>
Architect assertion 1.4.6: The architect asserts that:
A definition of Supra Complexes, their PrimePs, and their constituents' complexes.
</td>
</tr>
</table>

F.3.2 Predictive Thinking Pipeline 2: Resolve the Complicatedness pattern of the observed complexity.

The second pipeline in the predictive thinking process focuses on resolving the complicatedness pattern inherent within the observed complexity field. Building upon the holistic structural understanding established in Predictive Thinking Pipeline 1, this step delves deeper into the intricate web of interactions among the constituent elements of the problem domain. While Pipeline 1 provided a comprehensive, objective definition of the overarching structure of the problem domain complex, Pipeline 2 shifts attention toward uncovering and articulating the dynamic interdependencies and interactions that contribute to its complicatedness.

a) Step 2.1) Analyse Problem Interactions Using Actions Matrix

This step focuses on systematically mapping and analysing the interactions among key systems within the problem domain's complexity field. Using the Actions Matrix process, interactions are defined as binary relationships between source and sink systems, expressed in concise, action-based phrases. This process aims to clarify the dynamic interplay and potential dependencies among elements such as trains, adversarial drones, train tracks, fences, vegetation, and electric power lines. This step ensures that all relevant relationships are captured and contextualised by leveraging the General Systems Rules, particularly the principles of Complicatedness (General Rule C) and the superficial ArcMatrix (General Rule F).

The analysis begins with the list of coexisting elements identified in prior steps. It applies the predictive question, *What is the complete set of interactions among the defined complex within the problem domain's complexity?* The architect employs the Actions Matrix process to document these interactions systematically, ensuring that each connection is logically structured and captured in the format [source system][action][sink system]. For instances where the relationships appear ambiguous or counterintuitive, lateral Predictive Thinking Processes are applied to uncover less obvious connections. This often involves introducing auxiliary third-party elements that bridge the gap, thereby explaining intricate or indirect interactions.

The step’s completion is contingent upon identifying and defining all binary relationships, ensuring no interaction is overlooked. For example, interactions might include: [train][passes][train tracks], [adversarial drone][disrupts][surveillance cameras], or [vegetation][obstructs][fences]. By cataloguing N distinct interactions, this step creates a comprehensive picture of how the elements collectively contribute to the observed problem complexity.

The output of this analysis—a detailed list of interactions—serves as an essential artefact for understanding the problem's systemic nature and guiding subsequent design or intervention strategies. By explicitly defining the dynamic relationships among elements, this step lays the foundation for resolving complicatedness, mitigating risks, and ensuring that the architect's approach to the complexity field is robust, systematic, and actionable.

Table F.12 SECoT for Problem Interaction Analysis

SECoT Title	Problem Interaction Analysis
Input	Architect assertion 1.3.4 Architect assertion 1.4.4 Architect assertion 1.5.4
General Systems Rules	**General rule F:** Complicatedness definition. **Complicatedness:** the predictability of a given observation by the predictive architect's approach to minimising their epistemic uncertainty. It is the Impact of complexes' behaviours (events in observation) on some predictive architects' confidence in making decisions to manage, use or interact with the observed complexity field. **General rule G:** Actions Matrix.
Predictive Thinking Process	**Predictive question2.1.1:** What is the complete set of interactions among the defined complex within problem domain complexity? **Guiding prompt2.1.2:** Apply the Actions Matrix process and briefly describe the interactions among the list of contributing problematic coexisting elements as uncovered by Predictive Thinking Pipeline 1. Describe the interaction between the source and sink in a single verbal phrase. For every interaction where the interacting elements do not make sense, utilise the Lateral Predictive Thinking Process defined in section 5.3, and include at the end of the interaction the auxiliary third-party element where the intricate interactions made sense. Define the interaction using the following format : [source system][action][sink system]. **Note:** Some interactions are expected to be more challenging to understand their relevance. Those interactions are the information you have never thought about, forming the essence of prediction. You get to face the limit of your knowledge and thus the residual ignorance. Resolving those interactions requires lateral thinking. Thus, you need to note each interaction with: [Lateral perspective concerning...]. The way to solve it is by shifting your worldview and looking at the problem from a different systems perspective. One of them may be more likely to be impacted by this new unknown interaction. **Step completion criteria 2.1.3:** The step is considered complete when;

	All binary relationships have been identified among the identified list of problematic coexisting elements. Interactions are defined using the following format : [source system][action][sink system].
Output prediction	**Architect assertion:** The architect asserts that: List the complicatedness interactions of the problem domain complexity:

b) Step 2.2) Predict the contributing factors (unsafe situations or opportunities)

Knowing whether the architect is solving the right problem is the leading risk we are trying to mitigate in this process. In this step, the architect is challenged to make a judgement which situations are problematic or beneficial, thus helping the architect validating the relevance of their prior belief.

This step involves categorising the interactions identified in Step 2.1 into two distinct categories: **unsafe problematic situations** and **beneficial or non-problematic situations**. The purpose is to evaluate the dynamics of the identified interactions, elaborating on their aspects and potential impacts and systematically distinguishing between those that present risks and those that may offer opportunities or neutral outcomes. This categorisation forms a critical foundation for prioritising areas requiring deeper analysis and intervention in subsequent design and engineering stages.

To achieve this, the architect revisits the documented interactions from the Actions Matrix, systematically re-writing and re-evaluating each relationship in the context of its systemic effects. Unsafe problematic situations are interactions that pose clear risks, disrupt the system's intended functionality, or create emergent behaviours that could compromise safety, performance, or operational goals. These situations are flagged for additional scrutiny and further analysis in future steps. For instance, an interaction such as [adversarial drone][disrupts][surveillance system] might be classified as unsafe due to its implications for system integrity and security.

Conversely, beneficial or non-problematic situations are interactions that either contribute positively to the system's goals or present no apparent risks under the current context. These interactions are recognised as areas of stability or potential leverage for optimisation. For example, [train][passes][train tracks] could be considered beneficial, as it aligns with the primary purpose of the transportation system.

The step ensures that every interaction is elaborated upon to capture its anticipated effects, refining the architect's understanding of the problem domain. The outputs—categorised as unsafe situations or opportunities—provide a focused lens through which the system's complexity can be analysed. This categorisation not only aids in prioritising design efforts but also ensures that risks and opportunities are addressed holistically, supporting a structured and objective approach to resolving the system's complexities. This step can be captured in the following table format:

Table F.13 Problem characterisation template

Unsafe problematic situations	**Beneficial or non-problematic situations**
List of the problematic situations	List of the beneficial situations

c) Step 2.3) Design problem selection

This step involves re-evaluating the original problem to determine which sub-problems should be addressed further within the design process and which can be excluded. This re-evaluation involves classifying the outputs from Step 2.2 into two categories: *"To be solved"* for problems that fall within the design sphere of concern and *"To be dropped"* for those outside of it. The decision-making process includes providing justifications for these classifications and ensuring that the rationale behind the selection of design problems is transparent, systematic, and well-documented.

The goal of this step is to refine the focus of the design effort by prioritising problems that align with the stakeholders and the project's objectives, safety requirements, and operational scope. This involves assessing the significance of each interaction, its potential risks or benefits, and its relevance to the overall system purpose (PrimeP). Additionally, the step provides an opportunity for the design team to consider ethical implications, stakeholder concerns, and regulatory expectations, particularly when deciding to drop certain issues. This consideration supports the development of a trustworthiness case, ensuring that the design process is both robust and defensible.

For each interaction or concern, the step requires a concise elaboration or justification for the decision. For instance, an unsafe situation involving an adversarial drone crossing into a restricted train track zone might be labelled *"To be solved"* due to its critical impact on system safety. Conversely, a problem related to occasional electric sparks from a pantograph might be classified as *"To be dropped"* if the issue's frequency or impact is negligible.

This step creates a focused and actionable set of design priorities by systematically categorising and justifying the inclusion or exclusion of sub-problems. It also establishes a foundation for stakeholder communication, enabling the design team to provide clear reasoning to regulators, assurance teams, or other stakeholders on the scope and intent of the design decisions. Ultimately, this process enhances the alignment of the design effort with the system's goals while ensuring accountability and transparency in addressing the problem domain. The following is an example of defining problems:

Table F.14 Format of the design problem selection process

interaction	Potential concern	Decision	Elaboration or Justification
Interaction definition or an ID	Describe the nature of the concern	To be solved / to be dropped	Elaborate and justify your decision
n17: adversarial drone crosses track zone fence	The adversarial drone recognises and then crosses over the train track zone fence, which may lead to striking the train.	To be solved	One of the unsafe situations is when the adversarial drone successfully crosses into the zone. The design will focus on solving the issue of adversarial drones' unchallenged physical crossing over train track zones into train tracks zone open space.

F.3.3 Predictive Thinking Pipeline 3: Predict the Emergence of AIC Complexity Field for Detailed Operational Scenario Articulation

This stage involves a deeper analysis of the interactions identified in the previous pipelines, focusing on predicting the emergence of Appreciation, Influence, and Control (AIC) complicated behaviours within the problem domain. Building upon the refined problem set and interaction choices defined in Predictive Thinking Pipeline 2, this pipeline explores the intricate relationships and dynamic patterns that underlie each interaction. The objective is to uncover how these interactions manifest and may influence the system's behaviour across various scenarios, enabling a more detailed understanding of the problem domain's complexity.

a) Step 3.1) Model detailed AIC interactions scenarios for the problem domain

For every unsafe, problematic, or beneficial interaction, and considering the source complex's PrimeP, visualise the emergence of AIC complicated behaviour using an AIC modelling schema. In this case, any factor modelled must be written with a specific situation in the format of {adjective_name}. For example, if you define train as a factor, a situation needs to be mentioned with it, for instance {moving_train} or {roaming_adversarial_drone}. Being clear about a factor's dynamic or static situation is a critical thinking step to resolve the complicatedness of any complexity. AIC modelling schema models the dynamic and static situations of systems. Also, actions are to be modelled as verbal phrases and signed (-) if they carry an intended obstructive goal, (+) if they carry an intended supportive goal, or unsigned if they carry a neutral unintended impact.

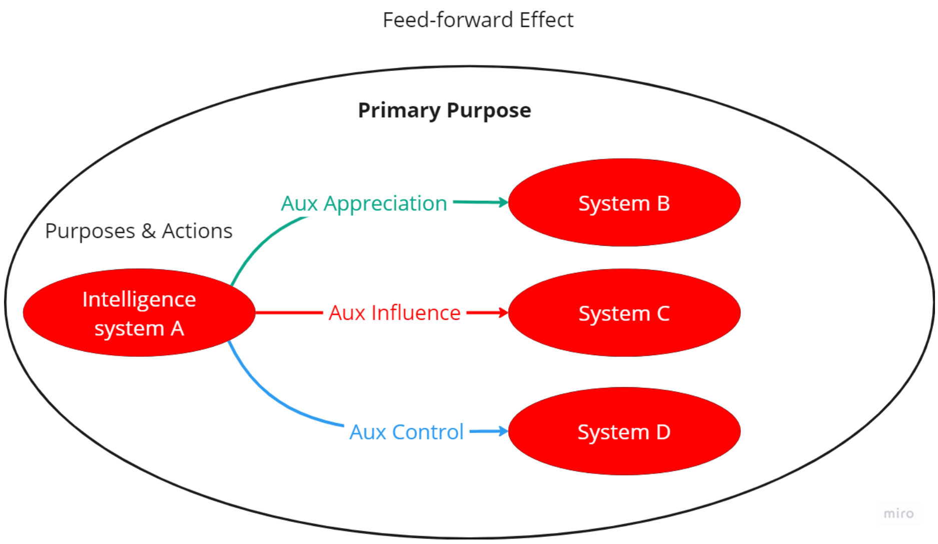


Figure F.2 Forward-Feed partial AIC-CoT

For example, consider adversarial drone PrimeP: disrupt trains operations. Our case study will consider only the Forward-Feed partial AIC CoT. We will choose the unsafe approach of a Train to an adversarial drone, which may lead to striking the train [derived from n6].

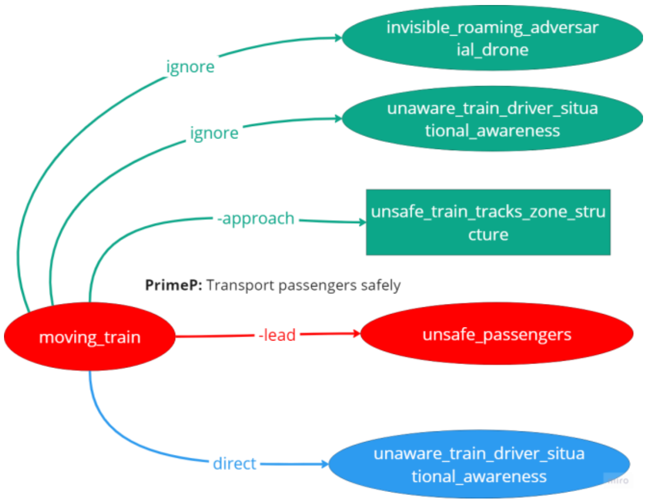


Figure F.3 Example Forward-Feed predictive thinking using AIC-CoT

b) Step 3.2) Predict the extended list of emergent AIC interactions scenarios

This step systematically predicts and categorises emergent interactions within the complexity field using the Appreciation, Influence, and Control (AIC) framework. It builds upon previously identified unsafe problematic situations and observed system actions, expanding the analysis to uncover auxiliary AIC interactions that may emerge dynamically under various scenarios. The objective is to construct a predictive model that encapsulates how system components interact in appreciation (situational awareness), influence (indirect effects), and control (direct governance).

The process begins by reviewing unsafe problematic situations and observed actions derived from earlier steps, using them as inputs to identify how these interactions align with or disrupt the system's supra-source Primary Purpose (PrimeP). Each interaction is detailed in terms of its role within the AIC framework: **Appreciation** interactions capture situational awareness, **Influence** interactions document indirect impacts and **Control** interactions represent direct interventions or governance. These relationships are documented in a structured table, with each interaction linked to specific elements and subsystems.

For instance, an unsafe situation involving an adversarial drone approaching train tracks might reveal appreciation interactions, such as the drone detecting visible track structures, influence interactions like the potential for derailment, and control interactions, such as managing the drone's flight dynamics or modifying physical barriers. Each interaction is described using a structured format, such as |{source}_[action]_{sink}|, to ensure clarity and traceability.

The output includes a detailed table of emergent AIC interactions, highlighting the system elements and their roles across appreciation, influence, and control. This step ensures that all potential behaviours, risks, and opportunities are captured and analysed, providing a robust foundation for system design and decision-making. By predicting these emergent interactions, the design team can pre-emptively address challenges, refine system architecture, and ensure alignment with the system's overall purpose. This structured approach enhances the ability to engineer resilient, adaptable systems capable of managing complexity and uncertainty effectively. We then capture the model using the following example SECoT:

Table F.15 SECoT for AIC complexity field prediction

SECoT title	AIC complexity field prediction	
Input	Unsafe Problematic Situations	
General Systems Rules	AIC Systems Approach Definitions	
Step 1: Unsafe Problematic Situations	Adversarial drones follow train tracks, which may lead to striking the train [derived from n12].	
Step 2: Observed System (obs)	**Step 3: Observed Action**	**Step 4: supra source Primary Purpose**

roaming_adversarial_drone, train_derailment	Roaming adversarial drone approach train	Adversarial Scheme PrimeP: Disrupt Train Network operations.
Step 5: Auxiliary Influence interaction	**Step 6: Auxiliary Control interaction**	**Step 7: Auxiliary Appreciation interaction**
|{roaming_adversarial_drone}_[-lead]_{train_derailment}|	|{roaming_adversarial_drone}_[+govern]_{roaming_adversarial_drone_flight_dynamics}|,|{roaming_adversarial_drone}_[-approach]_{train_tracks_zone_fence_structure}|	|{roaming_adversarial_drone}_[-detect]_{visitble_train_tracks_zone_visible_structure}|
Output prediction: Predicted Problem Domain Factors or Features (with repetition)		
The architect asserts that: Appreciation = ['roaming_adversarial_drone', 'visitble_train_tracks_zone_visible_structure'] Influence = ['roaming_adversarial_drone', 'train_derailment'] Control =['roaming_adversarial_drone', 'roaming_adversarial_drone_flight_dynamics', 'roaming_adversarial_drone', 'train_tracks_zone_fence_structure']		

c) Step 3.3) Collate factors / situations (step 8 in the table)

Collate all factors in {} and capture them in AIC lists. Include the following information types: Source, Sink, Supra Complex, Action and Goal. For example;

Appreciation = [Train_Network, moving_train, unsafe_train_tracks_zone_structure, train_transit_through_track_zone]

Influence = [moving_train, unsafe_unsecured_passengers_and_goods]

Control = [moving_train, unaware_train_driver_situational_awareness]

F.3.4 Predictive Thinking Pipeline 4: Predicting and Evaluating Problem Domain Factors and Assumptions

This pipeline focuses on systematically evaluating and defining the factors and assumptions that shape the problem domain's complexity, providing a foundation for analysis and decision-making. It involves a structured, multi-step process to organise, define, and critically examine the AIC factors and assumptions underpinning the whole complexity field behaviour. By performing this analysis, the design team can ensure a comprehensive understanding of the problem domain, identify gaps in knowledge, and refine the system's operational assumptions.

a) Step 4.1: Perform Most and Least Frequent Factor Evaluation

Evaluate AIC factors and list them without repetition from most frequent factor to least frequent. For example,

1. moving_train
2. unaware_train_driver_situational_awareness
3. train_transit_through_track_zone

The first step involves evaluating all identified AIC factors and organising them based on their frequency of occurrence across the problem domain. This step aims to prioritise factors by relevance and influence over our concern, according to how we view the problem, listing them from most to least frequent (most concern to lease concern). For instance, factors such as moving_train or train_transit_through_track_zone may occur frequently, highlighting their critical role in the system. This categorisation helps the design team focus on the most impactful elements while ensuring that less frequent but potentially critical factors are not overlooked.

b) Step 4.2: Define All Identified Problem Domain Factors

After identifying the factors, this step provides clear, concise definitions. The definitions should capture each factor's role, characteristics, and relevance within the problem domain. For example, moving_train might be defined as the continuous motion of a train along a predefined track. At the same time, unaware_train_driver_situational_awareness could refer to the limited perception of the train driver regarding unexpected events in the track zone. The goal is to understand and contextualise every factor within the system's complexity.

c) Step 4.3: Define the assumptions made about factors

This step evaluates the assumptions underlying the identified factors and their interactions. Each predicted emergent interaction is examined to articulate the assumptions influencing how the factors are understood and modelled. For instance, an interaction such as {roaming_adversarial_drone}_[-recognise]_{powered_powerlines_cables_structure_visual_

appearance} might assume that adversarial drones can recognise powerlines based on their visual features but do not account for conditions like random sparks or vibrations. Lateral Predictive Thinking Processes are applied to broaden the scope of assumptions, ensuring that implicit biases or overlooked variables are addressed.

Outcome

The output of this pipeline includes:

1. A prioritised list of factors ranked by frequency and relevance.
2. Detailed definitions of each factor, ensuring clarity and context.
3. A comprehensive list of assumptions for each interaction, providing transparency and traceability in the system's design rationale.

d) Step 4.4: Identify problematic Black Swan events[5]

As we now have a full picture of most factors involved in the problem complexity, we can make an expert judgment and pick those that we deem rare events in such complexity and relative to the operational environment. The thought process in this activity involves picking up factors and explaining why we think they are rare and impactful. We can also allow ourselves to think of different situations of the same factors, or if our predictive thought processes trigger us to imagine other scenarios. The main predictive question we need to ask ourselves here:

- What kind of rare, problematic, and impactful or not impactful scenario might arise from those factors? or,
- What rare and significant or insignificant events might happen?

A guiding prompt: You might find it helpful if you involve humour here. For example, think along the lines of “I tell you what, it would be funny if …” Then think of something strange but plausible and impactful, or even not impactful. Note we included both situations: impactful and non-impactful, because in the appreciation sphere, non-impactful and rarely concerning (by the architect) events are also appreciated as a possibility for concern.

The following attributes are required to capture problem level Black Swans:

- Predicted factors or interactions. For example, the visual complexity of plants leaves.
- Current architect concern level (the ratios mentioned in the analysis. For example, 1%.
- **Rarely concerning test:** the architect is prompted to assess whether a factor or an interaction rarely concerns them in their experience. For example, the architect had never encountered a harmful situation where the shape of a leaf was a primary concern.

[5] See section H.4.4.1 for an implemented example

- **Impact on ML perception reliability:** Would such a situation impact the reliability of the solution ML model?

F.4 SECoT_2: Hazards, Threats and Opportunities/Affordances Scenarios (HazTOPS) [6]

In traditional safety engineering intuition, HAZOP primarily focuses on identifying potential risks during control operations and within the complex boundary. In our application, we generalise this approach to discover emergent hazards to the operational environment and coming from the operational environment as well as the opportunities that might enhance an engineered complex's capability to fulfil its PrimeP. For instance, in the context of the Train Track Zone (TZ) Complicated problem, suppose the architect integrates an intelligent policing drone complex (Eagle Drone) into the police force (Supra Source).

The Eagle Drone aims to deny adversarial drone access into the train tracks zone. One aspect that the Eagle Drone must appreciate is the presence of wildlife (birds and insects). While wildlife around a TZ might pose risks to the Eagle Drone's operation (policing, it can also provide advantages. If adversarial drones use stealth mechanisms (such as forest-camouflaged fuselage) and trees for cover, the Eagle Drone can capitalise on the heightened sensitivity of wildlife, like birds, to drones. This reaction of the wildlife can serve as a cue for the Eagle Drone to pay extra attention in an area where those birds are abnormally reacting to detect hidden camouflaged drones, turning a potential hazard into an opportunity for better operability in certain scenarios.

An important aspect we identified in the traditional HAZOP study is using guide-words as creative thought steps to elicit hazards and operability issues. We view them as a Chain-of-Thought. We modified the established guidelines to fit within the broader scope of Complicated problems encompassing hard, soft, intelligent, and non-autonomous complexes using the AIC approach. We contextualise those guide words by generalising them as aspects of various behaviours and potential hazards related to contributing complexes, AIC goals and capabilities (actions and counteractions), supportive relationships leading to cooperation, or obstructive relationships leading to conflict.

We view discovering potential emergent scenarios as temporal, depending on the context being analysed. For example, suppose the architect aims to evaluate a current problem as-is and uncover any hidden emergent capability. In that case, they may think of “potential emergent scenarios where there is ...” If the architect wants to evaluate complexity in the future and

[6] See sections H.6.3 and I.6.3 for full implementation

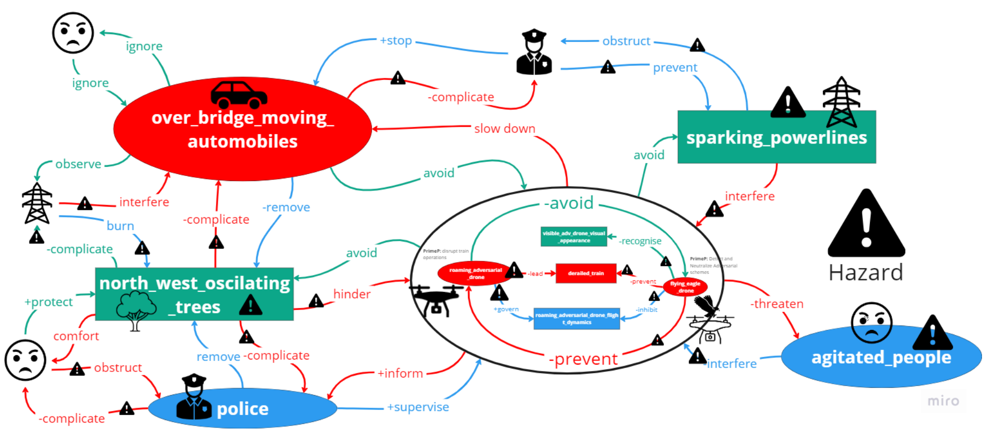
discover potential emergent scenarios after deploying the autonomous complexes, the architect may think about " there will be" scenarios.

There are three main components in HazTOPS:

- **Hazards:** sources of harm (safety context).
- **Threats:** sources of potential harm to system operability (security context)
- **Opportunities/Affordances:** This is an important aspect of the process as it prompts the architect to think of:
    - **Positive Affordances**: The architect can exploit situations that can be extracted from complexes to improve the quality of the engineered complexity. For example, the architect may use the presence of trees around train tracks as a natural shield or cover to ambush intruder adversarial drones.
    - **Negative Affordances**: These are potential threats associated with architect design decisions that other architects can exploit. For example, a regular ordered security patrolling of the Eagle Drone makes the drone's behaviour open for exploitation to develop counter technologies that impede the Eagle drone's effectiveness in delivering its PrimeP.

The general process for HazTOPS involves the following activities:

1. Ensure the problem domain's AIC schema is validated with stakeholders and ready for the HazTOPS stage.
2. Scope the potential safety and security challenges on the AIC schema of the problem domain.
    a. Add the following icon  for safety hazards,  for opportunities and  for potential security threats (cyber-attack) to every interaction on the model[7]. For example:



[7] The choice of icons is arbitrary for now. We do envisage that there would be a repository of standardised symbols or icons for systems engineering, just like those used for electrical components or in chemistry.

Figure F.4 Graphically scoping the hazards within the complexity field by placing hazard icons on target interaction.

b. Clearly define the nature of hazards, threats and opportunities in the scoped interactions. For example, a hazard coming from sparking_powerlines to eagle_drone in the form of electromagnetic interference on an Eagle Drone (the icon of a drone and an eagle).

3. Identify further potential complexity by utilising the following modified keywords: More, Part of, Less, Early, and Late. Follow SECoT_2 to derive the variety of potential deviations. Make sure you use all the words for each interaction, do not try to avoid guide words which you think might be irrelevant or do not make sense. Those interactions that feel this way are Black Swans with respect to your epistemic uncertainty.
4. The output of this process is a set of new HazTOPS scenarios written in the following format:

Table F.16 Hazards, Threats, and Opportunities/Affordances Scenarios Structure

HazTOPS Aspect	**Definition**
Hazards, Threats or Opportunities/ Affordances Scenario: Guide word	This section identifies scenarios as hazardous or opportunistic with a "Guide word" headline. e.g., HazTOPS Operating Scenario: more conflict with Local People
Operating Scenario Context: What will happen? Articulate a possible scenario.	This section provides a more detailed description of the scenario. For example, people decide to deploy their drones to monitor the Agent, leading to a swarm of Local drones that distract the Agent.
Hazard, Threat or Opportunity definition (consider used-complexes): Why would it happen? Articulate the causes that may lead to such an eventuality.	What exactly is the source of danger or opportunity? Consider using complexes or tools to identify hazards or opportunities in a scenario. e.g., an increased number of adversary drones piloted by local people in the vicinity.
Foreseeable Sequence of Events: How does it happen? Describe the	Specify the sequence of events that could occur due to a given scenario. Visualise and predict the chain of events

sequence of situations or events that could occur in the eventuality.	based on initial conditions. e.g., people utilise their back gardens adjacent to the Train Tracks Zone; People launch personal drones; Eagle Drone encounters additional adversarial drones.
Potential harm or benefit: So what? What would be of concern or interest? Predict the set of harms or affordances**.**	What is the ultimate outcome resulting from the danger or opportunity at hand? e.g., risk of drone collisions and decreased effectiveness of Eagle Drones.

Table F.15 shows a structured template to help architects systematically document and describe hazardous or opportunistic operating scenarios. Table F.6 also presents a structured template to help users report and describe hazardous or opportunistic operating scenarios systematically. The problem articulation process produces a set of potential factors. We use HazTOPS analysis to elaborate further on potential new complication scenarios that must be considered. The following is HazTOPS analysis guide-words (modified HAZOP guide-words) SECoT 2.

F.4.1 Predictive Guide Words for Potential Complicatedness

The HazTOPS keywords provide a structured approach to analysing complicated system interactions, offering insights into potential hazards and opportunities arising from changes in effectiveness, timing, or interactions. These keywords facilitate the systematic exploration of emergent dynamics within a complexity field, ensuring that all relevant aspects of a system's behaviour are considered. ***The main practice here with guidewords is that the architect is encouraged to think of scenarios inspired by all guidewords for every interaction***. If the scenario is deemed to be hard to think or imagine, that is good news. It is a Black Swan relative to your knowledge base.

This is only a generic definition of the keyword. We could make the process more rigorous by converting it into a SECoT; such an effort can be reserved for future research.

Table F.17 HazTOPS guidewords partial definition for superficial predictive thinking

Guide words	**Meaning**
More: increased uncertainty due to the involvement of more	**Definition**: Refers to increasing the number of interactions, elements, or complexities in a complex, which can lead to new opportunities or hazards.

things than initially expected.	**Example**: The introduction of more adversary drones into the airspace increases the likelihood of conflict and signal interference.
Part of: increased uncertainty due to unexpected involvement of Part of the source or sink	**Definition**: Refers to the involvement of only a portion of a complex or interaction, which can disrupt the complex's overall effectiveness. **Example**: Only part of the train's safety monitoring complex works, leading to incomplete track safety checks and potential hazards.
Less effectiveness: increased uncertainty due to unexpected reduction of some expected effectiveness	**Definition**: Refers to the reduced effectiveness of a complex's actions or interactions, which may weaken its overall function and stability. **Example**: The Eagle Drone's sensors become less effective in fog, causing it to misidentify the train track boundaries.
Before or after, Late or early: increased uncertainty due to unexpected temporal, special or systematic sequence changes.	**Definition**: Refers to timing issues where actions or interactions occur earlier or later than intended, unexpected change in temporal sequence or special positioning, potentially resulting in failures or missed opportunities. **Example**: The Eagle Drone detects an adversary drone too late, missing the chance to control or neutralise it in time.

Table F.16 outlines the definitions and applications of HazTOPS guidewords for superficial predictive thinking, focusing on key dimensions of complexity, interaction, and timing in problem scenarios. The **"More uncertainty"** guideword highlights how an increase in elements or interactions, such as introducing additional adversarial drones, can create opportunities or hazards by amplifying complexity. **"Part of source or sink"** addresses scenarios where only a portion of a complex system is functional, leading to risks like incomplete safety monitoring. The **"Less effectiveness"** guideword captures the impact of reduced functionality, such as diminished sensor performance under challenging conditions, which can weaken system reliability. Finally, **"Before or after"** and **"Late or early"** in the context of AIC timing emphasise the critical role of synchronisation, where delays or premature actions can result in missed opportunities or system failures, such as detecting threats too late to neutralise them effectively. This table provides a structured approach to understanding and analysing predictive complexities in systems.

F.5 SECoT_3: HazTOPS-driven Ordered AIC Requirements Elicitation Process

The HazTOPS-driven Requirements Elicitation process is a structured approach to identifying and defining requirements for Autonomous Systems (AS) by modelling interactions between agents and analysing potential hazards. The process integrates hazard identification with the appreciation, influence, and control (AIC) interactions framework to ensure that safety-critical complexes can adequately manage risks. We specifically mention Ordered AIC to refer to the following of the normal AIC intuition. Refer to section:

F.5.1 Predictive Thinking Pipeline 1: Introducing Autonomous systems into Forward-Feed complexity

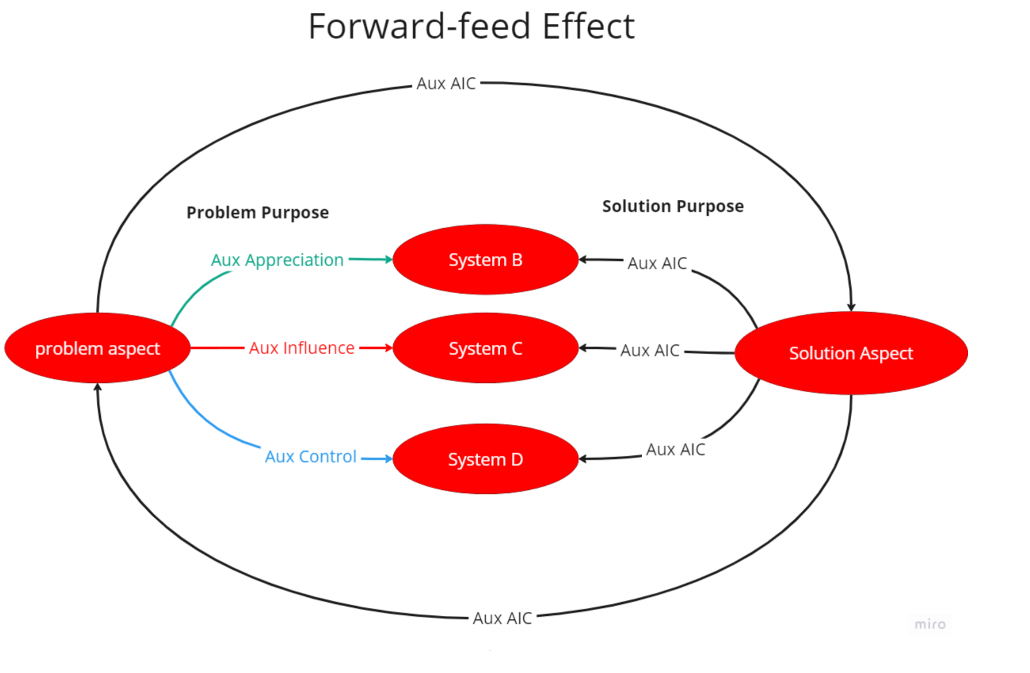
In this step, we derive the desired agent AIC interactions that fulfil the architect's intents and stakeholders' needs. The Predictive Thinking Pipeline strategy is defined using the following SECoT. We will use the following interaction example: Adversarial drone avoids vegetation around the fence, which may lead to a train derailment.

a) Step 1.1) Introduce the solution into the mix.

Model the AIC Schema and introduce the solution into the mix of the problem situation. We select an influence interactions-based assumption as a starting point. For example, {roaming_adversarial_drone}_[-lead]_{train_derailment}. This influence interaction belongs to the unsafe behaviour of: 2.4.Adversarial drone avoids vegetation around the fence, which may lead to a train crash. Furthermore, Include the Architect's Intent (mission).

Table F.17a solution introduction SECoT into problem complexity

Input	Interaction
General Systems Rules	**General rule G:** Emergence of AIC Complicated behaviour. **General rule H:** Forward-Feed Effect AIC modelling schema.

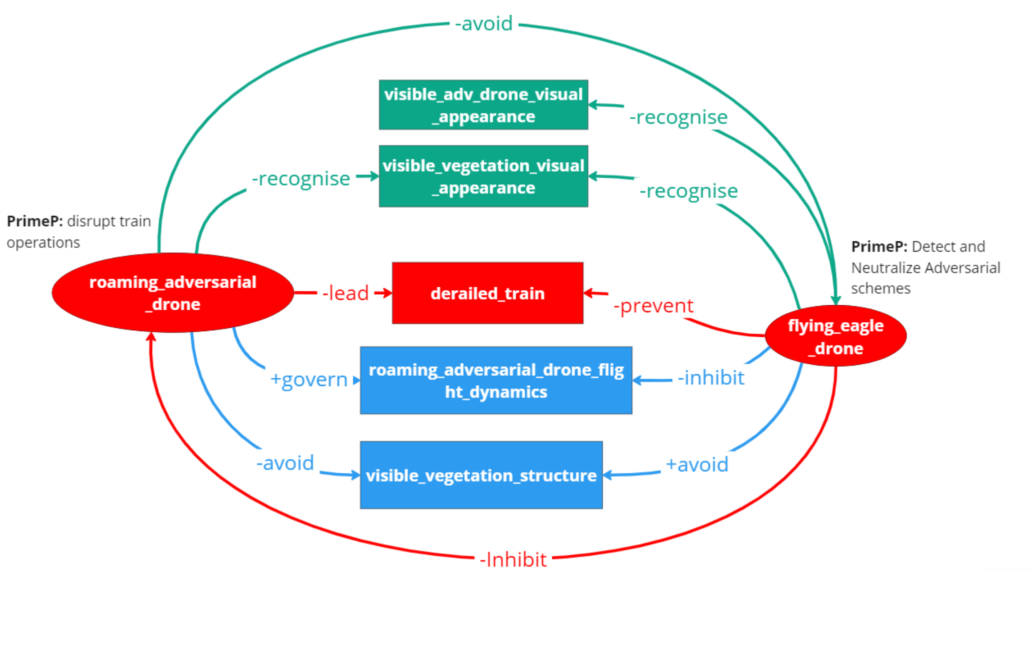
<table>
<tr><td></td><td>
</td></tr>
<tr><td>Predictive Thinking Process</td><td>Predictive question: What is the main problematic situation that needs to be influenced to achieve the influence? What situation needs to be controlled? What situation needs to be appreciated to ensure that the autonomous systems guarantee control?

Guiding prompt: Define the autonomous systems Forward-Feed AIC interactions with the rest of the problem.

Map the autonomous systems to all the parts of the problematic situation and define the actions and effect types for each interaction.
1. Start with the counter-influence intra-reaction to the main problematic situation.
2. Then, define which part of the problem needs to be controlled to achieve the influence.
3. Then, define which part of the problem needs to be appreciated such that the control can be achieved.

Step completion criteria: The step is considered complete;
• All Forward-Feed AIC binary relationships have been modelled between the autonomous systems and the problematic situation.</td></tr>
<tr><td>Output prediction</td><td>Architect assertion: The architect asserts that:

Include the solution AIC model. For example;</td></tr>
</table>



b) Step 1.2) Formalise the AIC interactions

Define the set of interactions between the sink and the source using AIC structured interaction format of Complex's Interaction formula (section 6.2.5):

|{source situation}_[+,- or no sign, AIC-action]_{sink situation}|,

Written in the following grammar:

|{adjective+noun}_[verbal phrase]_{adjective+noun}|[8]

For example,

|{flying_police_robot}_[learns humans' visual profiles]_{distressed_people}|

Capture the output in the following table:

Table F.18b AIC Interactions capture for system-level requirements

Source: source complex for example {eagle _drone}	
Output Behaviour	**Input Behaviour that impacts the emergence of Output Behaviour**
In: influence interaction	**An:** Appreciative interactions.

[8] See section 4.

	Cn: Control interactions
I2: |{flying_eagle _drone}_[-inhibit]_{roaming_ adversarial_drone}|	**A1:** |{flying_eagle_drone}_[-recognise]_{visible_vegetation_visual_appearance}| **A3:** |{flying_eagle_drone}_[-recognise]_{ visible_adv_drone_visual_appearance}|
	C1: |{flying_eagle _drone}_[-inhibit]_{roaming_ adversarial_drone_flight_dynamics}| **C2:** |{flying_eagle_drone}_[-avoid]_{visible_vegetation_structure}|

F.5.2 Predictive Thinking Pipeline 2: Designing the affecting Backward-Feed complexity

AIC Systems theory tells us that all systems impact and are impacted by the direct and indirect operational environment. The latter general rule helps us to predict that, at any given issue of complexity, there are several systems which are impacting that issue, even if it appears there is only one dominate impactful system. Thus, they must be taken into consideration. Therefore, it is essential to note that every node and edge in the AIC model impacts and is impacted by the systems around their direct and indirect operational environment. We need to define (in the context of the AIC complexity) every AIC relationship between the aspects of the model and the operational environment. To objectively analyse the situation, we must highlight the regions of interest in the image.

In this thought process, we will consider external factors that both affect and are affected by the complex behaviour. We will also consider systems that Appreciate, Influence, or control the Eagle Drone AIC behaviour.

a) Step 2.1) Visualise the operational design domain environment

In this step, we would take a realistic photo of the operational domain and imprint the AIC interaction model to help us refine our mental model. Based on the visualisation, we then construct the operational model context.

The operational domain context model distinguishes between **immediate factors** directly affecting the Eagle Drone's operational decisions and remote factors indirectly impacting its operation. These two levels of influence are described below:

1. Remote Affecting & Affected Problem Complex

These elements indirectly influence the operational domain or are less likely to be directly encountered by the Eagle Drone. However, they can still affect its mission objectives and overall situational awareness.

2. Immediate Affecting & Affected Problem Complex

These elements **directly impact or are impacted by the Eagle Drone's operations**. They represent the immediate environmental factors the drone must continuously monitor, respond to, and navigate.

b) Step 2.2) Backward-Feed Complicated Behaviour definition

In this step, we will extend the forward-feed model with the factors within the operational domain complexity by identifying which of the complexes identified in the real-world problem complexity. To do so, we may need to abstract the Forward-Feed model for ease of modelling. Then, we model the Backward-Feed AIC interactions with the abstracted model:

- What complex must appreciate the interaction? How should the source of the interaction react?
- What complex influences the interaction? How should the source of the interaction react?
- What complex controls the interaction? How should the source of the interaction react?

c) Step 2.3) Comprehensively appreciate the complicatedness of the problem domain

To comprehensively define the complexity of the above complexity abstraction, you may use the Actions Matrix process to explain all the actions and their effect among the complexes. Note that some actions may not make sense. Those interactions represent the lack of knowledge that the architect has about the problem domain (the residual ignorance) hence, the process proved itself to be useful in reducing the epistemic uncertainty.

We represent complexes in icons to better visualise the problem and aid with enhancing imagination. For example, the Eagle Robot complexity involves the following complex of complexes:

Table F.19 complexity field icon definition

Situation	Icon
flying_eagle_drone	

roaming_adversarial_drone	

And to resolve the complicatedness (epistemic uncertainty) we use the Actions Matrix. For example:

Table F.20 Eagle Robot complexity field action matrix

					
		-inhibit	+inform	avoid	distract
	-avoid		-avoid	gets in-between	Roam over
	+supervise	-capture		?	?
	hinder	hide	?		?
	observe	observe	?	?	

Question marks denote a harder-to-predict complicatedness since it is hard to comprehend how those complexes affect each other. Those questionable interactions are the Residual Ignorance.

d) Step 2.4) Complete the extended view of the AIC mental model schema

The Actions Matrix above defines how the scenario's complexity is resolved. Now, we will transfer all the knowledge to the step's final output, a comprehensive AIC mental model of the problem.

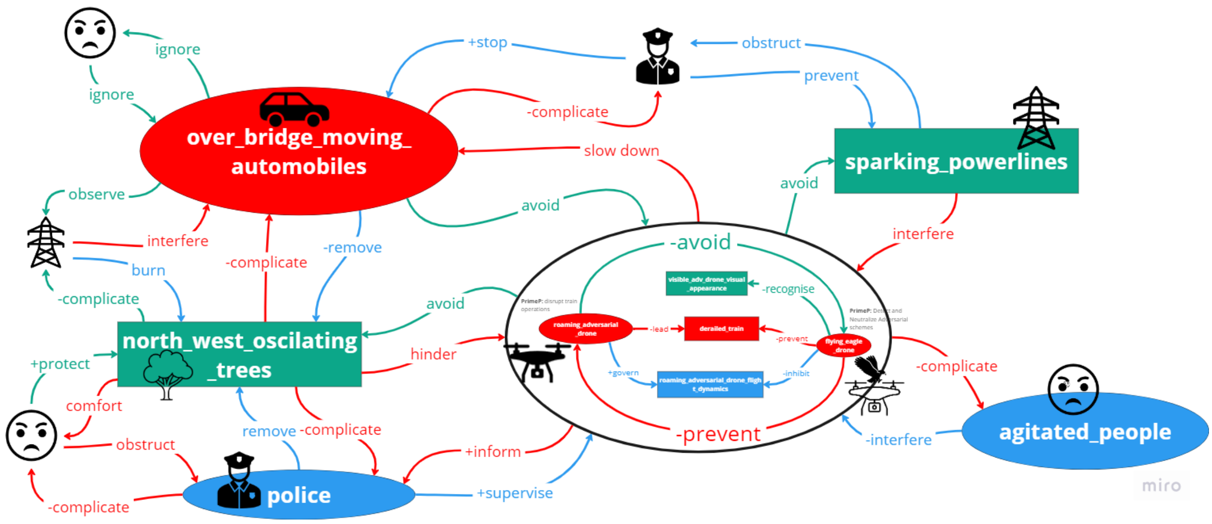


Figure F.5 Example complexity filed related to Eagle Robot case study

e) Step 2.5) Capture the AIC interactions

When examining complexity from any complex's point of view, the main objective is to examine how the rest of the complexity affects the complex's chances (probability) of manifesting its PrimeP. We would extend Table F.26 to include the residual AIC interactions (Residual Ignorance is mitigated, and the epistemic uncertainty of the compounded uncertainty has been reduced).

F.5.3 Predictive Thinking Pipeline 3: Hazards, Threats and Opportunities/Affordances Scenarios (HazTOPS) Analysis

We apply the HazTOPS outlined in section F.4. The following is the application of the process:

a) Step 1) Scope the HazTOPS context domain

Scope the potential safety and security challenges on the AIC schema of the problem domain.

Add the following icon  for safety hazards,  for opportunities and  for potential security threats (cyber-attacks) to every interaction on the mode. See Figures F.7 and F.8

For example,

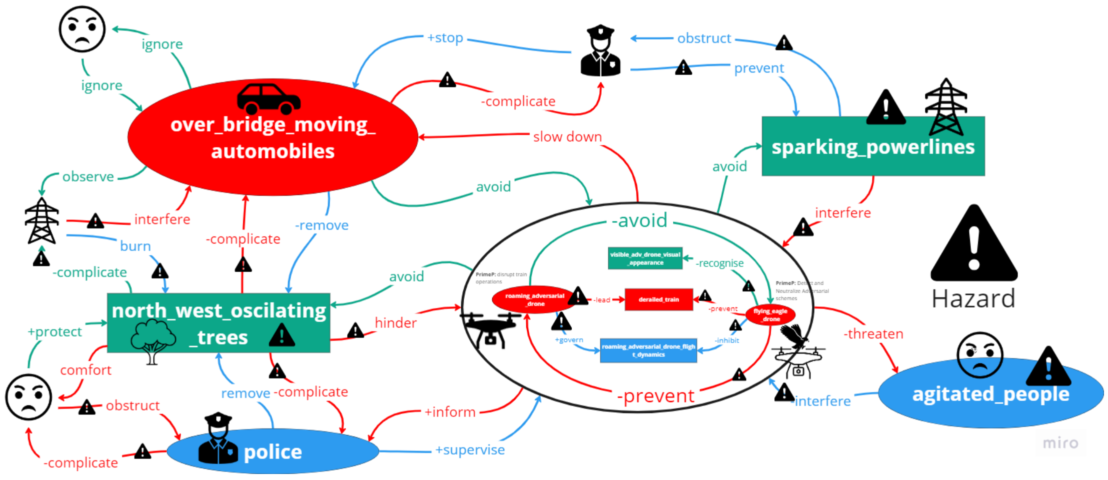


Figure F.6 Sources of Hazards Complexity Field example

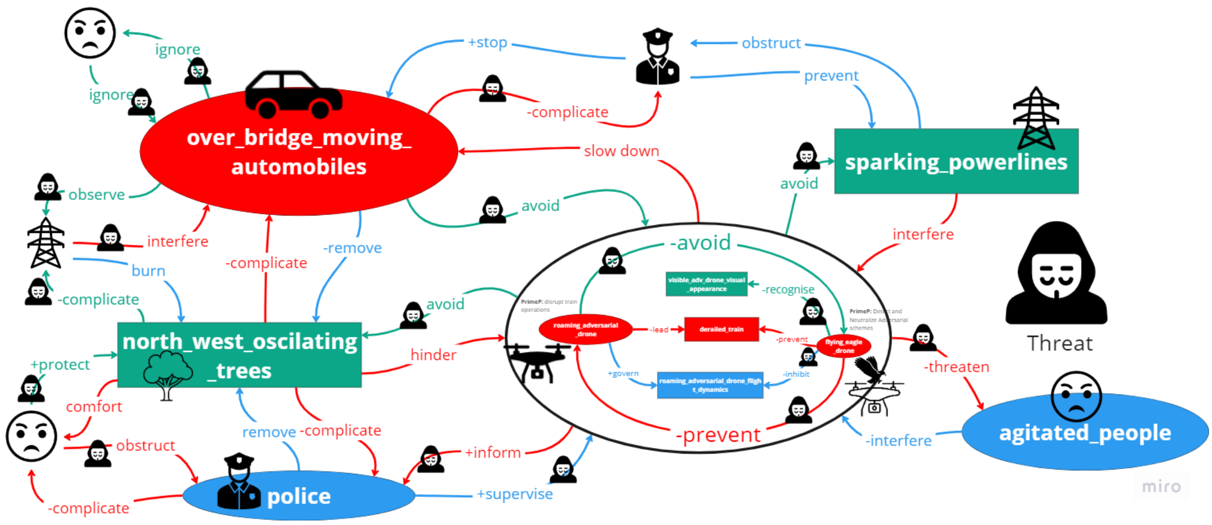


Figure F.7 Sources of Security Threats Complexity Field example

b) Step 2) Formulate the scoped HazTOPS interactions

Formulate the AIC complexity field interactions using the format Complex's Interaction in section 4.4.3 and capture the AIC interactions using AIC-Actions in section F.1.2 and as per table 6.10 in section 6.4.3.2. See the table below:

Source: source system	
Output Behaviour	**Input Behaviour that impacts the emergence of Output Behaviour**
Influence interaction	**Appreciative interaction**

	Control interaction

c) **Step 3) Apply predictive potential complications guide words.**

Identify further potential complexity by utilising the following modified keywords: More, Part of, Less, Early, and Late. Follow SECoT_2 to derive the variety of potential deviations. Then, include a risk and surprise analysis for each HazTOPS scenario.

F.5.4 Predictive Thinking Pipeline 4: Elicitate Ordered AIC System-Level Requirements

The **Predictive Thinking Pipeline 4** process focuses on systematically modelling hazardous scenarios, refining agent-environment interactions, and eliciting system-level requirements to ensure robust performance in complex, dynamic environments. This process emphasises breaking down high-level influence relationships into granular Appreciation, Influence, and Control (AIC) actions, identifying gaps or hazards in system functionality, and defining mitigation strategies to address them. By iteratively refining these interactions, the process ensures a harmonious and effective system design. This process results in:

1. **Refined Scenario Models**: Detailed representations of agent-environment interactions, identifying potential hazards and opportunities.
2. **Mitigation Strategies**: Clear actions to resolve hazards and enhance system harmony.
3. **System Requirements**: Comprehensive safety and performance requirements, ensuring the agent's actions align with its Primary Purpose (PrimeP) and operational goals.

a) **Step 1)** Model the Complex of Interest Operating Scenario Context

The first step involves creating a high-level model of the operating scenario, highlighting the agent's emergent capabilities and unresolved interactions with the surrounding environment. The system's AIC relationships are abstracted and visually modelled to identify influence-related gaps or inefficiencies. For instance, unresolved influence relationships—such as adversarial interference or local community objections—are refined into actionable Appreciation and Control relationships, ensuring clarity and tractability. These refinements enable the emergence of capabilities like secure navigation or patrol. For example:

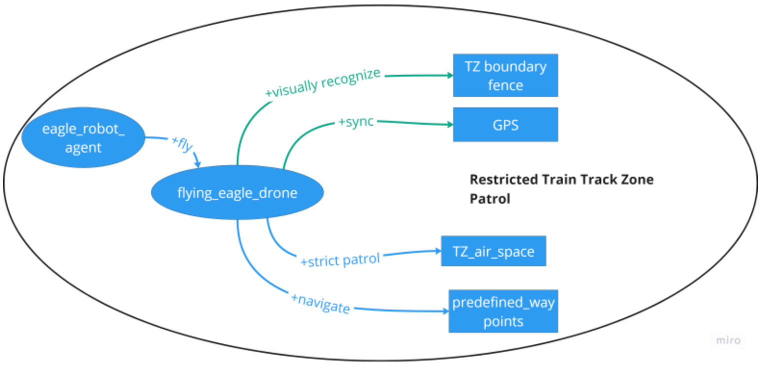


Figure F.8 Restricted TZ security boundary model (taken from figure 9.22)

b) Step 2) Model hazards mitigation ordered-AIC complexity field

This step models potential hazards, such as obstructive actions from external systems (e.g., laser interference by local people) for example:

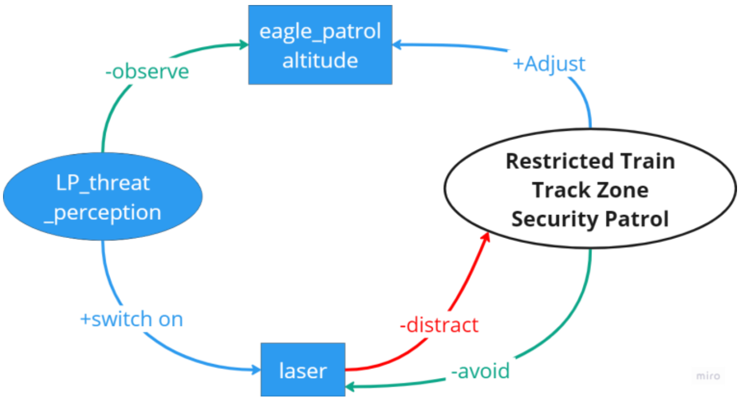


Figure F.9 Further refined relationship between Agent Capability and Local People(LP)

Then, defines counteractions or supportive actions to mitigate these threats. The modelling process breaks down complicated relationships into their constituent AIC elements, considering:

1. **Source Situation**: The agent's action initiating the interaction.
2. **Interaction Dynamics**: The flow and impact of actions within the complexity field.
3. **Sink Situation**: The target action or behaviour influenced by the source.
4. **Mitigation Strategies**: Steps to resolve hazards by introducing refined Appreciation and Control relationships. For example, mitigating laser interference involves enabling the agent to detect and avoid laser attacks while harmonising relationships with local people through adjusted patrol altitude.

The refined interactions are visually represented in updated models, reflecting changes in complexity and highlighting the transition from influence-based relationships to actionable appreciation and control strategies. For example:

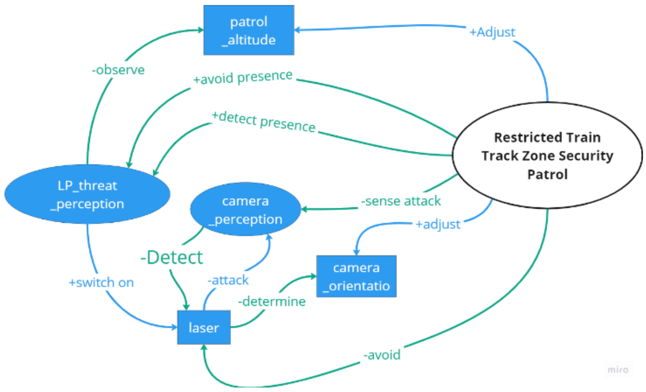


Figure F.10 Final refinement of the Local People and the Agent relationship resolves the restricted security emergent capability (taken from Figure 9.26).

The End of **step 2** would be collating the all-agent Appreciation and Control actions of the final refinement, considering any other actions from supportive systems to enable agents to counter actions further and be more effective. For example, in the case of the Agent aiming to avoid laser attacks, it needs supportive action from the Train Network and Police to mandate no laser usage by Local People against patrolling drones. To do so, we use the following HazTOPS Resolution Table:

Table F.21 HazTOPS Resolution Table template

HazTOPS Title	Title of the HazTOPS process	
Agent target Output interactions	**Agent Input interactions**	**Actions from supportive systems**
In: influence interaction	**An:** Appreciative interactions.	**AIC**
	Cn: Control interactions	

For example:

Table F.22 Example HazTOPS Resolution Table

HazTOPS Title	agitated_people use lasers	
Agent target Output interactions	**Agent Input interactions**	**Actions from supportive systems**
<u>Target Influence:</u> **I1:** {flying_eagle_drone}_[+strict_patrol]_{track_zone_ boundary}	**<u>Appreciation:</u>** **A4:** {flying_eagle_drone}_[-senses]_{laser_attack}	**A3:** {monitoring_police}_[+dictates]_{predefined_eagle_drone_patrol_area_ strategy}
	<u>Control:</u>	

	C2: {flying_eagle_drone}_[+adjust]_{camera_orientation}	

c) **Step 3) Ordered-AIC-based Mitigating System or Safety Requirements Derivation (Safe Operating Concept)**

All identified AIC actions are converted into system-level requirements in the final step. These requirements align the agent's capabilities with its operational goals, ensuring safe and effective performance. Each requirement is formulated using the following structure:

1. **Given**: The people direct laser attacks on the eagle drone perception,
2. **In Order To**: minimise their perception of the eagle drone threat.
3. **Then**: The Eagle Drone shall sense the laser attack (A4), and The Eagle Drone shall adjust its camera orientation upon detecting a laser (C2),
4. **In Order To**: secure the track zone within its boundary.

The architect can use the following table format to capture the process**:**

Table F.23 Safe Operating Concept Development

AC interaction	**Mitigating Safety or Systems Requirements (Safe Operating Concept)**
Identify the AC interaction of concerns.	Define the Safe Operating Concept
C2: {flying_eagle_drone}_[+adjust] _{camera_orientation} **A4:** {flying_eagle_drone}_[-senses]_{laser_attack}	**Safety Requirement 1: Eagle Drone recognises and avoids laser attacks.** **Given**: The people direct laser attacks on the eagle drone perception, **In Order To**: minimise their perception of the eagle drone threat. **Then**: The Eagle Drone shall sense the laser attack (A4), and The Eagle Drone shall adjust its camera orientation upon detecting a laser (C2), **In Order To**: secure the track zone within its boundary.

d) Step 4: Extended Concrete Safe Operating Concept and ML Safety Training Concept

This stage allows the architect to elaborate further on how to satisfy the operational environment and systems safety requirements, with more specific solution requirements that pertain to each unique interaction in the system. This process is not just a task; it represents a critical exercise in computational thinking, where the architect can apply concrete and reductionist processologies. These techniques transform abstract interactions into well-defined subsystem specifications, essential for clarity and successful implementation. In this step, the architect may also derive some of the training requirements for the ML component. Using the following format:

The AS ML component shall be trained to ...

It is imperative to acknowledge that there are equivalent testing requirements for every machine learning training requirement. We refrain from differentiating between Training and Testing and opt to designate only the term “training" for simplicity in description. When developing training datasets, practitioners typically determine a portion of said datasets to serve as testing. The assumption here is that the datasets produced by this stage account for a common and regular understanding (intuitive) of the complexity field. For example, we wouldn’t consider counterintuitive examples. However, this is not a hard rule but rather a soft guidance. The counterintuitive examples (such as a moon appearing like a traffic light in an image) would be part of the black swan scenario derivation stage. Those examples should be used to validate the black swan's performance. Furthermore, we distinguish the training syllabus into safety training or non-safety training. Depending on whether we are deriving training requirements from a safety or non-safety higher level requirements.

A general training requirement is necessary to bridge the conceptual gap between what needs to be covered in training, testing, and validation datasets and what the safety requirement dictates. From the training concept, we then derive a set of Dataset Requirements, using the following format:

The AS ML component [Training/Testing/Black Swan Training Subset] Dataset shall provide the trainee model with a valuable minimum variety[9] of ...

To illustrate the practical application of this approach, we will take an abstract AIC interaction or a system-level requirement (from step 3) and meticulously break it down into more concrete specifications. This breakdown is carried out using a process known as the '5-Whats-&-How,' as detailed in section F.8. This process helps identify the fundamental components and

[9] If you are wondering what the “Valuable minimum variety” constraint is, the concept is fully explained in section E.6.3.1.

relationships within the system, ensuring that each aspect of the interaction is thoroughly understood and accurately represented in the specifications. To capture the Safety Training Requirements:

Table F.24 ML Safety Requirements Derivation table

Safety requirement (Safe Operating Concept)	**ML Safety Training Requirements (Training Concept)**
Define the safety requirement.	Define associated ML training requirements.
Given: The people direct laser attacks on the eagle drone perception In Order To minimise their perception of the eagle drone threat. **Then**, The Eagle Drone shall sense the laser attack (A4), and The Eagle Drone shall adjust its camera orientation upon detecting a laser (C2) **In Order To** secure the track zone within its boundary.	**ML Safety Training Requirement 2:** The Eagle Drone's ML component shall be trained to recognise various laser attacks on its perception.

The CuneiForm characterisation process requires some preparations involving accepting an input safety requirement over perception training and testing datasets by articulating the set of at least one pictorial situation. For example, we will use the following:

> **ML Safety Training Requirement (Training concept):** The Eagle Drone's ML component shall be trained to identify various adversarial drones, especially unusual or stealthy ones.

From which we derive the Dataset Requirement:

> **ML Development Dataset Requirement:** The Eagle Drone's ML component Training Dataset shall provide the trainee model with a valuable minimum variety of adversarial drones, especially unusual or stealthy ones.

Multiple different CuneiForms variations can be produced to satisfy a requirement for datasets. In this section, we will focus on producing one CuneiForm. More variations can be produced by changing the number of TOIs and types of characteristics. To understand the constrain of "A valuable minimum variety of" see section E.6.3.

Non-Safety ML Training Requirements:

Sometimes, training requirements may not stem from safety requirements but from system requirements that are not necessarily safety issues but desired performance expectations. Such training requirements will be denoted as merely "ML Training Requirements" which means they are not safety-driven. For example, in our AVOIDDS case study, we identified two system requirements in Table I.26 (Appendix I). Those requirements are then used to derive ML Training Requirements, which are not safety issues but merely functional needs.

F.6 Multi-Purpose Operational Design Domain Definition (ODD)[10]

The definition of an Operational Design Domain (ODD) is a critical step in the systems engineering lifecycle for safety-critical autonomous systems. The ODD specifies the conditions under which the autonomous system is expected to operate safely and effectively, serving as the foundation for system design, validation, and operational assurance. Below, we outline a generic process for defining an ODD, followed by an example-based explanation and a structured table to formalise the definition. The architect can decide on the granularity of the Definition. However, we recommend that safety-critical applications use all the categories mentioned in Appendix A. Appendix B defines the operational environment uncertainty grade.

F.6.1 Step 1: Identify the System of Interest

The first step is defining the autonomous system the ODD applies. This includes understanding the purpose and context of its deployment. For example, suppose the system is an autonomous drone like the Eagle Robot. In that case, the focus might be on surveillance and operational support within a specific environment, such as a train track zone.

F.6.2 Step 2: Define the Solution Operational Space

The operational space refers to the physical and functional domain where the system will be deployed. This step involves identifying the geographical region, infrastructure, and the boundaries of the system's operational zone. For instance, the operational space for the Eagle Robot could be "Train Track Zone in the UK," where specific challenges such as restricted airspace and proximity to critical infrastructure are considered.

[10] See sections H.7 and I.7 for implementation example

F.6.3 Step 3: Determine the ODD Uncertainty Grade

Assign an uncertainty grade to represent the complexity and variability of the operational environment. Safety-critical applications typically range from Grade A (ideal and highly controlled conditions) to Grade E (highest uncertainty conditions). For example, in a Grade A scenario, the Eagle Robot might operate in clear weather conditions with minimal environmental variability.

F.6.4 Step 4: Specify Development Purpose

Articulate the intent behind defining the ODD, such as system training, testing, or operational deployment. For the Eagle Robot, the ODD may be designed to train the system for obstacle detection and test its stability under specific environmental conditions.

F.6.5 Step 5: Characterise the Environment

Define the environmental parameters that the system is designed to handle, including weather, lighting, and natural phenomena. This step ensures the system is only exposed to conditions it has been validated to handle. For instance, for the Eagle Robot, the environment might include:

- Lighting.
- Weather.
- Time of the Year: Seasons-specific environmental characteristics.
- Landscapes type variety definition.
- Geographical region-specific natural phenomena.
- Time of the Day.
- Perceived Horizon Attitude.
- Sun sphere positioning.
- Moon sphere positioning.
- Specialised zones features.

F.6.6 Step 6: Formulate the ODD in a Structured Table

Consolidate the ODD definition into a structured table that provides clarity and traceability for all stakeholders. The table should include:

- **System of Interest:** The autonomous system being defined.
- **Solution Operational Space:** The geographical and functional boundaries.
- **ODD Uncertainty Grade:** The level of complexity in the operational environment.
- **Development Purpose:** The primary intent of defining the ODD.

- **Environmental Characteristics:** Detailed environmental conditions, such as lighting, weather, and visibility.

Table F.25 Example ODD format

System of interest		Eagle Robot
Solution Operational Space		Train Track Zone in the UK
ODD Uncertainty Grade		Grade A
Development purpose		Training and Testing
Environment Characteristics		**Ideal natural environment** system for perception
Natural Lighting Conditions		Sunny
Weather Conditions	**Precipitation** mm/h	0
	Wind km/h	<5
	Humidity %	50
	Visibility km	>8
	Cloud Cover	Clear or Blue Sky (0/8 oktas)
	Snow mm/12 hrs	0
	Pollen	No pollen
	Sand	Neither sand nor dust storms
	Temperature	15-25 degrees Celsius
	Sunshine Duration	12 hours
Time of the Year: Seasons-specific environmental characteristics		1 type of season
Landscapes type variety definition		1 type of Landscapes
Geographical region-specific natural phenomena		0 or 1 type of Infrastructure
Time of the Day		Midday (Noon)
Perceived Horizon Attitude		1 type

Sun sphere positioning	1
Moon sphere positioning	0
Specialised zones features	1 feature

F.7 AIC perspective shift

AIC perspective shift is a lateral thinking technique for Black Swan Scenarios discovery. A Black Swan Scenario is a lesser expected, harder-to-foresee, whose impact cannot be ignored situation in the real world. However, an intelligent system is expected to make the right decisions, for example, the food delivery robot in a situation like the one described above. In the literature, the term "Edge Cases" [4] is commonly used to describe the harder-to-predict scenarios. However, we believe such char1acterisation is an inappropriate adaptation of a concept well understood in deterministic software development, where maximum and minimum input boundaries are identifiable. For non-deterministic complexes such as ML-based perception, there is no recognisable boundary of inputs for pictorial datasets. For example, how can we specify a "human" in a pictorial sense to clearly define maximum or minimum edge cases? Using the term "Black Swans, which are Hard-to-Foresee Emergent Scenarios", is more appropriate, for example, when facial recognition datasets did not account for people wearing COVID-19 protective masks.

Predicting Black Swan scenarios is crucial for robust design and operational resilience in complicated architecture.[11]Common hazards and risk analysis techniques may help developers discover unsafe scenarios. However, more needs to be done regarding intelligent systems design since their emergence is exponentially more unpredictable than that of non-intelligent systems. The AIC (Appreciation, Influence, Control) perspective shift technique, as applied within the AIC General Systems approach (GSApp), enables architects to predict potential harder-to-predict unsafe scenarios by shifting perspectives within known events. This process triggers the architect's predictive thinking by reimagining events from different angles within the AIC framework.

[11] By "resilience," we mean the robustness of the solution architecture to cope with the change in the complexity field, thus remaining relevant and effective in delivering the desired PrimeP. In simple terms, resistance to change of requirements.

F.7.1 Predictive Disordered AIC Perspective Shifts Processes

The AIC perspective shift is a lateral Predictive Thinking Process that leverages a dual-dimensional approach within the AIC framework, facilitating a re-evaluation of known events or interactions from alternative perspectives. This technique comprises four types of shifts: altering the AIC type of an interaction, reversing the direction of the interaction, altering the type of effect for an action and altering all kinds together.

The term "Disordered" relates to the fact that by the end of the requirements modelling of the complex, the architect would have generated an AIC-ordered model (assumed AIC intuitive model) of the complexity field. In this step, we challenge the architect's imagination (predictive capability) to consider scenarios by altering the underlying perspective (considering the potential of a complementary, confusing, complex or counterintuitive model). This allows for discovering Black Swan scenarios (concerning the complexity field of the solution) or black-sawn events. This process is an objective process for lateral thinking since it takes the thought process out from what is deemed to be intuitive into a counter-intuitive perspective, thus supporting the architect in utilising lateral thinking.

1. Predictive Perspective Shift by Altering Interaction's AIC Type

This dimension entails modifying the AIC type of interaction within a complex, transitioning an element's role from appreciation to influence or control or vice versa. This allows architects to explore how changes like interaction affect complicated behaviour and outcomes. The shift involves redefining the functional purpose of an element in a scenario, such as changing an advisory tool into an authoritative controller. This transformation prompts a re-evaluation of complicated dynamics and user interaction patterns.

Example

Consider an aircraft avoidance recommendation complex (source) initially designed to influence the pilot's decision-making process (sink) in a supportive manner. Applying the perspective shift, the complex's interaction is reimagined as controlling, potentially obstructing the effectiveness of the pilot's intuition role in the autonomous decision-making process.

Black Swan Scenario prediction

- **Initial scenario (Influence)**: The relationship between the recommendation complex and the pilot is primarily influential and supportive. Pilots might initially perceive the complex as somewhat unreliable, preferring to rely on their intuition.

The perspective is then can be shifted from influence to control to view this complex as a controlling complex:

- **Post-shift perspective scenario (Control)**: If we alter the perspective, we can envision the recommendation complex directly controlling the pilot's behaviours, particularly in high-stress situations where the pilot may depend entirely on the complex's directives, potentially to the detriment of safe flying practices. Such a situation can be considered as a surprising, Black Swan Scenario.

2. Predictive Perspective Shift by Reversing Interaction Direction

This approach reverses the flow of interaction, transforming the source of an action into its recipient (sink), and vice versa. This reversal challenges conventional flow dynamics within the complex, prompting new insights into potential complicated behaviours and vulnerabilities. Reversing the interaction direction can reveal vulnerabilities or strengths in the complex by showing how changes in the flow of information or control can lead to different outcomes.

Example

Weather conditions (source) influence an aircraft collision avoidance recommendation complex (sink). The complex must appreciate the weather when attempting to control the types of recommendations it makes while influencing the pilot’s decision-making process to avoid a potential collision.

Black Swan Scenario prediction

- **Initial scenario (Influence)**: An initial interaction between the weather conditions (appreciated sink) and an aircraft collision avoidance recommendation complex (appreciating source), with the recommendation complex being influenced by environmental conditions.

By reversing this interaction, the aircraft can be imagined as an influencer of the environment:

- **Post-shift perspective scenario (reversed influence)**: If we alter the influence direction, we have a situation where the aircraft influences the environment. This would be counterintuitive and surprising; what kind of scenario could the aircraft be viewed as having influenced some aspect of the environment? This might involve the aircraft's complexes modifying environmental data inputs due to a cyber-attack or a bug, which could lead to erroneous sensor data readings being fed into navigational or operational complexes. In the context of Eagle Robot (Figure 5.14)**:**

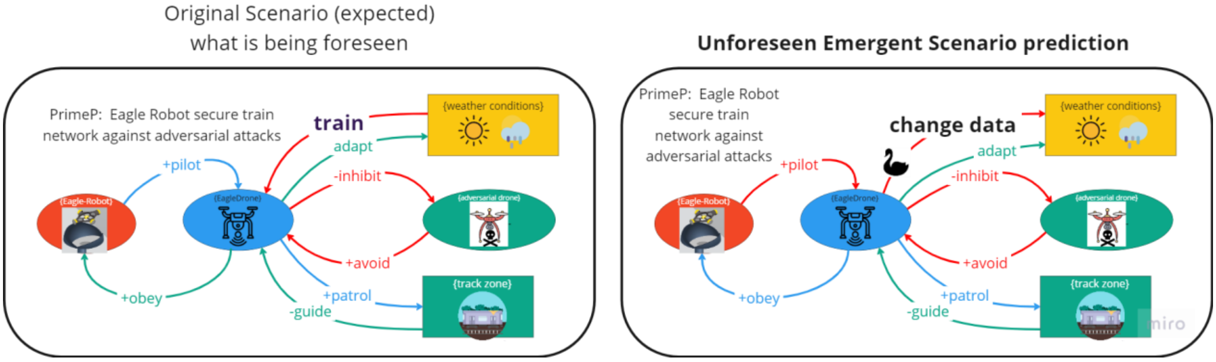


Figure **Error! No text of specified style in document.**.11 Reverse interaction perspective shift model

3. Predictive Perspective Shifts by altering types of effect [Supportive, Obstructive, Neutral]

The perspective shift by altering the type of effect involves changing the impact that the actions of one complex (source) have on another complex (sink) in terms of being supportive, obstructive, or neutral. This shift allows architects to evaluate the adaptability and resilience of complexes under different operational conditions by envisioning how variations in the effect type could alter behaviours and outcomes. This analysis is crucial for understanding a complex's potential vulnerabilities or strengths that may not be evident under normal conditions.

Example

In its standard operational mode, an aircraft collision avoidance complex enhances safety by providing timely warnings and manoeuvre suggestions to avoid potential aerial conflicts. The complex continuously monitors the airspace for other aircraft and uses algorithms to predict potential collision paths. It then alerts pilots with specific avoidance manoeuvres.

Black Swan Scenario prediction

- **Initial scenario (supportive effect)**: The complex supports the pilot by augmenting their situational awareness and decision-making capabilities, helping to maintain safe distances from other aircraft and navigate congested airspaces effectively.

- **Post-shift perspective scenario (alternative effects)**:

The perspective is then can be shifted by re-imagining the type of effects, changing from supportive to obstructive or neutral:

Obstructive Effect

1. **Post-shift perspective**: By altering the perspective to an obstructive effect, we can reimagine the collision avoidance complex as temporarily hindering the pilot's ability to execute certain manoeuvres.
2. **Operational Scenario**: During a high-density traffic situation at an airport, the complex might excessively restrict pilot manoeuvre options to prevent potential collisions. This might force pilots to follow less optimal paths or delay landings, which could disrupt flight schedules and increase fuel consumption.
3. **Impact**: Here, the complex, usually supportive, now acts as an obstruction to efficient flying, prioritising safety over efficiency and convenience.

Neutral Effect

1. **Post-shift perspective**: Shifting to a neutral effect, the complex could be imagined as non-responsive or minimally interactive, neither aiding nor hindering the pilot actively.
2. **Operational Scenario**: Consider a scenario where the complex enters a diagnostic mode during flight, ceases to provide active collision warnings, but continues monitoring airspace without issuing commands.
3. **Impact**: In this mode, the complex neither supports nor obstructs the pilot, leaving them to rely entirely on their training and other navigational aids. This could be particularly challenging during low visibility conditions or in highly trafficked airspaces.

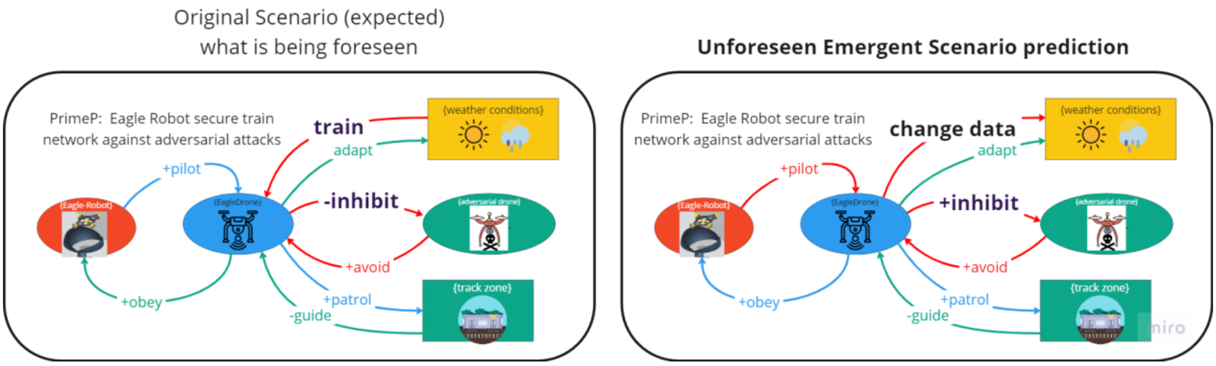


Figure **Error! No text of specified style in document.**.12 Altering types of effect perspective shift model

4. Predictive Multi-types Shifts by Altering AIC and Effect Types, as well as Interaction Direction

This comprehensive shift combines altering the AIC type and effect type and reversing the direction of interaction, providing a robust process for evaluating complicated interactions. This multidimensional shift enables exploring complicated scenarios where changes in complicated operational dynamics might occur. By simultaneously shifting the AIC type and interaction direction, architects can simulate complicated interactions that might not be apparent under normal analysis conditions.

Example

Initially, the aircraft avoidance recommendation complex influences pilot decision-making:

- **Initial scenario**: The complex provides recommendations, and the pilot uses these to make informed decisions.

The scenario is then reimagined with both the AIC type and interaction direction altered:

- **Post-shift perspective scenario**: The pilot controls the recommendation complex's sensitivity settings, directly manipulating its operational parameters and recommendations. This thought process may prompt us to question what kind of scenario in which the pilot side controls recommendation parameters. Does this mean we need to give the pilot such a level of control? Also, it may prompt us to question whether there may be a cyber-attack that exploits pilot access to the recommendation complex.

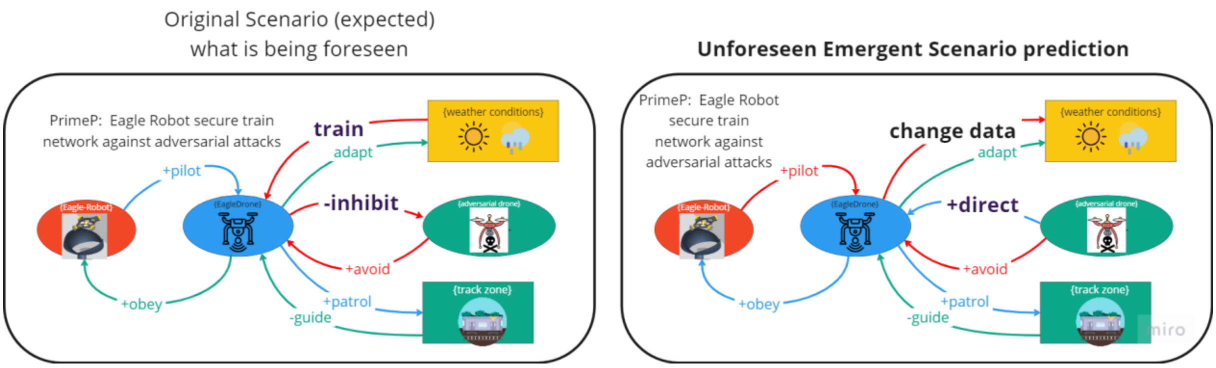


Figure **Error! No text of specified style in document.**.13 Multi-types Shifts model

F.7.2 Predictive Disordered AIC Perspective Shift SECoT_4

To apply the concepts above, we defined AIC perspective shifts SECoT in Table F.20. The process outlined in Table F.20 involves a process aimed at predicting Black Swan scenarios from anticipated complicated interactions by shifting perspectives based on AIC factors. Given a description of a phenomenon and assuming a pre-determined AIC factorisation of relationships, these relationships are expected to change due to evolving complexity. The process involves four steps:

1) Assessing what happens if the interaction flow and effect type remain constant while altering the AIC type of purpose and action;

2) considering the impact of the interaction purpose and action type stay the same but the interaction flow direction reverses;

3) exploring the outcome of the interaction purpose type, action type, source action, and interaction flow direction remain constant, but the action effect changes.

4) evaluating the scenario when all AIC factors (interaction purpose, action type, source action, and interaction flow direction) change.

Each step is guided by specific prompts to redefine actions and scenarios within the shifted context, ultimately generating four Black Swan scenarios for each complicated phenomenon described. The following chain of thought describes what to do for each step:

Table **Error! No text of specified style in document.**.26 Disordered AIC Perspective Shifts Chain-of-Thought SECoT_4

SECoT Title	Disordered AIC perspective shift
SECoT Primary Purpose	To assist with predicting unexpected, Black Swan scenarios from anticipated complexes interactions
SECoT Input	An AIC model of some situations.
General Systems Rule	Given a pre-determined AIC factorisation of a relationship, it is expected that a certain AIC-formulated relationship will change unexpectedly due to a change in complexity over time.
Predictive Predictive Thinking Process	**Step 1) General complexes rule**: given a source or sink, it is possible that over time and with the change of complexity, the AIC goals are altered, leading to a new situation of complicatedness. **Predictive question:** What would happen if the interaction flow and effect type remained the same, but the goal's AIC type and action type were altered in the future? **Guiding Prompt:** Review the AIC dynamics of the observed complexes and alter the nature of goal and action. Then, define an appropriate action to bear alternative AIC types and describe a scenario in the shifted context. You may consider disordered AIC timing (before or after). Consider section E.2 (AIC Timing). **Completion criteria:** The step is considered complete when a scenario demonstrating an alternative AIC goal is detailed.
	Step 2) General complexes rule: Given a source or sink, it is possible that over time and with a change of complexity, the direction of interaction is revered, leading to a different situation of complicatedness. **Predictive question:** What would happen if the interaction goal and action type remained the same but the interaction flow direction reversed in the future? **Guiding Prompt:** Review the situation of the observed complex and swap

	only the interaction direction, making the source the sink and the sink the source of the new potential interaction. Then, define an appropriate new action to bear the initial AIC types but with a reversed interaction flow direction and describe a scenario in the shifted context. You may consider disordered AIC timing (before or after). Consider section 6.4 (AIC Timing). **Completion criteria:** the step is considered complete when a scenario is generated such that the original flow of interaction is clearly reversed.
	Step 3) General complexes rule: Given a source or sink, it is possible that over time and with a change of complexity, it is possible that the action effect type (supportive, obstructive, neutral) changes in typeness, leading to a different situation of complicatedness. **Predictive question:** What would happen if the interaction goal type, action type, source action, and interaction flow direction were the same, but the action effect changes in nature in the future? **Guiding Prompt:** Review the situation of the observed complex and only change the effect type (supportive, obstructive, or neutral). Then, describe a scenario in which the nature of the source action is different. **Completion criteria:** The step is considered complete when a generated scenario reflects a clear change of action type.
	Step 4) General complexes rule: Given a source or sink, it is possible that over time and with a change of complexity, all AIC factors change, leading to a different situation of complicatedness. **Predictive question:** What would happen if all AIC factors' interaction purpose, action type, source action, and interaction flow direction were changed in the future? **Guiding Prompt:** Review the situation of the observed complex and describe a scenario whereby the purpose and action types (appreciation, influence, control), as well as the effect type (supportive, obstructive, or neutral), are all changed. Reverse the direction of the interaction and the action itself too. You may consider disordered AIC timing (before or after). Consider section E.2 (AIC Timing). **Completion criteria:** The step is considered completed when
SECoT Output	Four different Black Swan scenarios for every given complex phenomenon.

F.7.3 General predictive AIC perspective shift process:

To implement the AIC perspective shift, we need to perform the following process:

Step 1) Define the interactions required to predict a potential emergence.

Step 2) Define the interaction's current AIC factors using the AIC Factorisation process.

Step 3) Perform the perspective shift using AIC perspective shift SECoT. Any scenario predicted from a shifting perspective is a black swan event. If a scenario cannot be imagined during analysis, it should be recorded as "unknown unknown" .

Step 4) Predict Black Swan scenarios (black swan scenario).

In this step, the architect may expand further on the prediction using the 4WnH (section F.8) or 5-whys process.

Step 5) Define mitigating strategy or safety requirements (safety and training concept).

Define safety and training requirements from the predicted scenarios to mitigate the black swan scenarios.

The above steps can be captured in the following table:

Table **Error! No text of specified style in document.**.27 AIC-type perspective shift SECoT format

Interaction	AIC interaction		
AIC factors	**Pre-shift perspective**	**Post-shift perspective**	**Black Swan Scenario (black swan scenario)**
Source	Complex	Complex	Post-shift perspective scenario
Sink	Complex	Complex	
Supra Source	Source Supra Complex	Source Supra Complex	
PrimeP	Source Supra Complex Primary Purpose	Source Supra Complex Primary Purpose	
Source's Goal	Source's goal	Source's goal	
Source's Goal type	AIC	AIC	
Source's Action	Action	Action	
Source's Action type	AIC	AIC	
Source action effect on sink	Obstructive, Supportive or Neutral	Obstructive, Supportive or Neutral	

For example, suppose we have the Eagle Drone tasked to patrol the track zone. To perform a type perspective shift, we must fix the pre-shift perspective of all AIC factors and alter the AIC purpose and action type from appreciation to control. Then, we will reimagine a situation where the perception system controls the laser attack's behaviour. We can think of "reflecting," considering that a reflective material could coat the camera lens. This realisation allows us to predict a potential Black Swan Scenario.

Table **Error! No text of specified style in document.**.28 AIC-type perspective shift example

Interaction	The camera-complex perception detects laser attack light		
AIC factors	**Pre-shift perspective**	**Post-shift perspective**	**Black Swan Scenario**
Source	**Camera perception**	**Camera perception**	The camera perception could potentially lead the drone complex to behave as if it seeks to reflect the laser attack beam by orienting the camera lens (coated with a reflective coating) to reflect the laser attack away.
Sink	**Laser attack**	**Laser attack**	
Supra Source	Police Force	Police Force	
PrimeP	Ensure the safety of the Train Network	Ensure the safety of the Train Network	
Source's Goal	Show of security presence	Show of security presence	
Source's Goal type	***Appreciation***	***Control***	
Action	**Detect**	**Reflect**	
Action type	***Appreciative***	***Controlling***	
Source action effect on sink	**Obstructive**	**Obstructive**	

Here is a step-by-step explanation of how this perspective shift leads to the derivation of a Black Swan scenario:

Purpose Type: From Appreciation to Control

- **Pre-shift perspective**: Initially, the purpose of the camera complex's interaction with the laser is appreciation. This means the complex acknowledges the laser's presence but cannot influence or alter the laser attack impact or whatever is controlling it.
- **Post-shift perspective**: The purpose shifts to control, indicating a proactive approach, a presence of capability, an emergent one where it is not expected, to managing the laser's impact. Instead of simply detecting the laser, the complex aims to mitigate or eliminate its effect.

Action Type: From Appreciative to Controlling

- **Pre-shift perspective**: The original action is appreciative, confined to detecting or recognising the laser attack.
- **Post-shift perspective**: The action becomes controlling, involving direct intervention to alter the situation of the laser attack, specifically through reflecting it.

Action: From Detect to Reflect

- **Pre-shift perspective**: The fundamental action is to detect the laser light, a passive form of perception. Detecting carries an appreciation perspective with it.
- **Post-shift perspective**: This changes to "reflect", as in reflecting light, which is an active and direct process of manipulating the laser light to prevent it from achieving its intended effect. Reflecting action carries a perspective of control.

F.8 Disordered-AIC Black Swan Scenarios Predictions

This section outlines a generic system engineering process for analysing and predicting **Black Swan scenarios** using the AIC (Appreciation, Influence, Control) framework, emphasising addressing disordered timing and emergent complexities. The process integrates AIC perspective shifts, predictive analysis, and validation strategies to mitigate unforeseen risks. Black Swan scenarios are rare, high-impact events not typically considered during conventional modelling, making this stage critical for validating and enhancing the robustness of autonomous systems.

F.8.1 Step 1: Define the Interactions

The first step involves defining key interactions within the complexity field to identify relationships that may lead to emergent behaviours. Architects can pinpoint potential vulnerabilities or Black Swan triggers by selecting relationships of interest. For example, consider the interaction between the **flying Eagle Drone** and **adversarial drone shapes**:

- **Interaction Example:** The Eagle Drone recognises adversarial drone shapes in its operational environment to detect and respond to intrusions.

To enrich the analysis, interactions are extended by including binary relationships and the reactions of each system component, which illustrates the interconnected influences among systems.

F.8.2 Step 2: Define the ArcMatrix

This step applies the **Deep AIC Matrix** to break down the interactions into detailed Appreciation, Influence, and Control factors, creating a granular view of each system's goals, actions, and effects. For example:

Table F.29 The ArcMatrix for the adversarial drone complex and the Eagle drone

	Flying_eagle_drone	**Adv_drone_shapes**
Flying_eagle_drone		**Supra Source:** Train Network. **PrimeP:** Safely transport people and goods. **Goal:** detect adversarial drone presence. **Goal type:** Appreciation. **Action:** recognise adversarial drone shape. **Action type:** Appreciation. **Effect:** Obstructive.
Adv_drone_shapes	**Supra Source:** Adversarial Scheme. **PrimeP:** Disrupt Train Network operations. **Goal:** avoid detection by Eagle Drone. **Goal type:** influence. **Action:** obscure Eagle Drone perception. **Action type:** Influence. **Effect:** Obstructive.	

F.8.3 Step 3: Perform the Perspective Shift

The **AIC perspective shift** introduces alternative goals and actions to simulate potential Black Swan events. Using the **SECoT_3 thought process**, the interaction between the Eagle Drone and the adversarial drone shapes evolves as follows:

1. **Initial Perspective**:
   - **Goal**: Influence – Adversarial drones obscure the Eagle Drone's perception.
   - **Action**: Avoid detection (influence-type action).
2. **Shifted Perspective**:
   - **Goal**: Control – Adversarial drones seek to neutralise the Eagle Drone's monitoring ability.
   - **Action**: Camouflage (control-type action), leveraging advanced technology to blend with environmental elements.

This shift demonstrates how complexity evolves, requiring systems to adapt dynamically. **Figure 9.35** illustrates the transition from influence to control.

F.8.4 Step 4: Predict Black Swan Scenarios

At this stage, the focus is on identifying Harder-to-Foresee Emergent Scenarios or the Black Swan scenarios and elaborating their sequences. For instance:

- **Black Swan 1**: The adversarial drone deploys advanced camouflage to blend with the environment (e.g., sky, trees, or infrastructure), rendering it nearly invisible.
- **Black Swan 2**: The Eagle Drone's monitoring capabilities degrade due to the adversarial drone's increasingly sophisticated tactics, resulting in undetected incursions.

You may also try using **4WnH** (section F.10). These scenarios highlight how evolving tactics can introduce unprecedented challenges, requiring new countermeasures. Finally, the **rationale for each predicted** scenario is required in order to justify the feasibility of black swan scenarios and added to the safety assurance argumentation. We capture the derived information using the following table:

Table F.30 Black Swan Scenarios Capture

Black Swan Scenario	**Rationale for prediction: it is possible that ...**
Define black swan scenario	Explain why it is feasible and why it should be part of the design
Black Swan 1: The adversarial drone may shift tactics, using advanced camouflage technology.	With modern developments in adaptive camouflage, such as optical cloaking and AI-driven pattern adaptation, adversarial drones could blend seamlessly into environmental

	elements like the sky, trees, or train infrastructure.

F.8.5 Step 5: Safety Requirements for Black Swan Scenarios

Safety requirements are derived to address Black Swan scenarios to guide system design and ML Black Swan Training Subset. The process for addressing Black Swan scenarios begins with **defining the scenario**, identifying potential rare and high-impact events, their characteristics, and how they disrupt the system. For instance, an adversarial drone using advanced camouflage to blend into environmental elements like trees or infrastructure compromises the monitoring and security capabilities of systems like the Eagle Drone.

Safety and Systems requirements derivation:

Next, **safety objectives and goals** are established to maintain system functionality and mitigate the identified risk. For example, the Eagle Drone would aim to detect and neutralise camouflaged adversarial drones. From these objectives, **safety or system requirements** are derived, specifying the conditions, expected outcomes, and actionable responses the system must perform, such as implementing machine learning algorithms to dynamically identify and track camouflaged drones on the ground.

Safety ML Development Requirement derivation:

Training the Eagle Drone to recognise laser attacks is part of satisfying safety requirements. Thus, we need to specify a training requirement for the ML model. In this case, this is a safety training requirement since it mitigates a safety hazard.

> **Safety ML Development Requirement 1:** The Eagle Drone's ML component shall be trained to recognise laser attacks and perform avoidance manoeuvres.

Capture the analysis results using the following table. Since this particular requirement is not in the NL Safety Development, we decided to add it to the training concept. The architect may choose to have the requirement only for the Black Swan Training Subset process. In this case, the architect may define "Black Swan Training Subset" instead of "training."

Table F.31 Black Swan Scenarios Mitigation Process

Black Swan Scenario	**Mitigating Safety or systems requirements (Safe Operating Concept)**	**Safety ML Development Requirements (Training Concept)**

Black Swan 1: The adversarial drone may shift tactics, using advanced camouflage technology.	**Safety Requirement 5:** Eagle Drone track and trace camouflaged adversarial drones on the ground.	**ML Safety Training Requirement 4: Recognise Camouflaged drones on the ground** The Eagle Drone's ML component shall be trained to detect, classify, and track adversarial drones using camouflage tactics on the ground by recognising subtle movement patterns, environmental inconsistencies, and spectral differences using AI-driven computer vision models.

F.9 AIC-based 4-Whats-&-How refinements

The **4-Whats and How concrete refinement strategy** is designed to systematically elicit concrete sub-system-level requirements by analysing influence relationships within a complex field in a hierarchical manner. The process reuse **SHARCS** [4] hierarchical abstraction technique. This process focuses on breaking down abstract interactions into actionable Appreciation, Influence, and Control (AIC) relationships, enabling the identification of operational requirements that ensure system robustness and adaptability. By iteratively exploring "What" needs to be influenced, "how" to be controlled, and "what" to appreciate, the process ensures a comprehensive and traceable foundation for requirement elicitation.

The main objective is to model the complexity field driving emergent behaviours and define detailed system-level requirements by decomposing influence relationships into their constituent elements of control and appreciation. This process allows for precisely identifying operational parameters, interactions, and supporting mechanisms critical to achieving the system's Primary Purpose (PrimeP). Although the process aims to decompose influence interactions, it can also be used to decompose any control or appreciation interactions. The process is generic and can be adapted according to the need.

F.9.1 4-Whats-&-How (4WnH) Process Overview

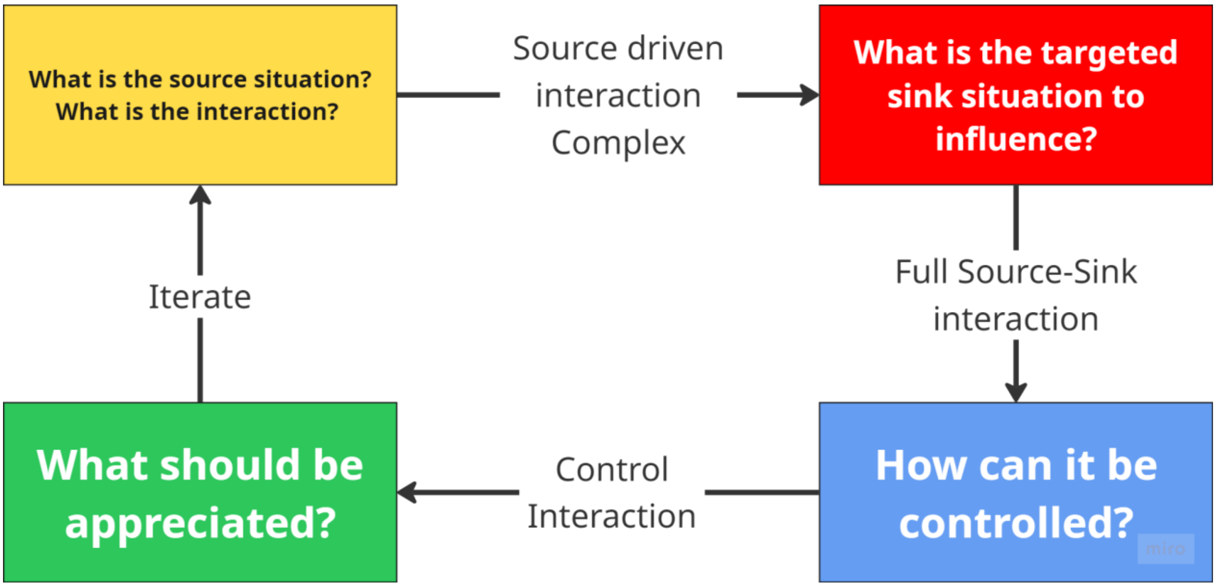


Figure **Error! No text of specified style in document.**.14 4WnH thought process

The **4-Whats and How Analysis** (Figure 5.11) involves iteratively answering the following key questions for every influence relationship:

1. **What is the source situation?** Identify the initiating source complex or component responsible for the interaction.
2. **What is the interaction?** Define the action being exerted by the source on the sink (e.g., regulation, adjustment, or modulation).
3. **What is the targeted sink situation to influence?** Specify the target system, action, or behaviour that needs to be influenced.
4. **How can it be controlled?** Determine the specific control actions required to manage or direct the interaction effectively.
5. **What should be appreciated?** Identify the appreciation mechanisms (e.g., sensors, feedback systems) necessary for situational awareness and adaptive responses.

Repeat the process 5 times for every discovered control complex. Capture the process using the following table:

Table **Error! No text of specified style in document.**.32 4-Whats-&-How process with an example

Source situation	**interaction**	**Target Sink situation (What to influence)**	**How to control it? (control)**	**What to appreciate? (appreciation)**

Eagle_drone	+flying	low_patrolling _altitude	+Regulate adaptive_ Flight_controller	+Visually scanning visible_terrain, GPS

The process also includes an AIC hierarchical schema modelling to assist with visualising the whole fractal of subsystem components. The schema provides an interactive and layered understanding of the system's behaviour, highlighting interdependencies between influence, control, and appreciation.

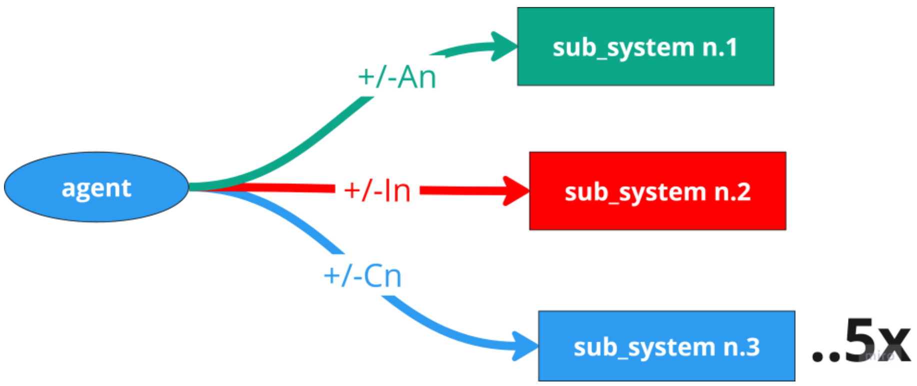


Figure **Error! No text of specified style in document.**.15 AIC Hierarchical decomposition of interactions

F.9.2 Application: Case Study of the Eagle Drone

To illustrate this process, we apply the **4-Whats and How Analysis** to the Eagle Drone system tasked with low-altitude patrolling. The analysis captures the relationships between the drone's components, its operational environment, and supporting systems:

1. **Source Situation**: The Eagle Drone initiates the interaction by flying at low patrolling altitude.
    - **What is the Interaction?**: The drone must influence its adaptive flight controller to regulate altitude and maintain patrolling efficiency.
    - **Target Sink Situation to influence**: The adaptive flight controller governs the altitude and navigational accuracy required for secure operations.
2. **How to Control It?**
    - **Control Actions**:
        - Regulate variable and fixed altitude through adaptive flight control.
        - Adjust continuous rotor speed to maintain stability and lift.
    - **Mechanisms**:

        - Dynamic flight controllers modulate proportional thrust and lift forces.

3. **What to Appreciate?**
    - **Appreciation Mechanisms**:
        - Visually scan the visible terrain and GPS for navigation.
        - Monitor barometric sensors for altitude precision.
    - **Feedback Systems**:
        - Continuous rotor speed and thrust sensors provide real-time input for adaptive adjustments.

For the next refinement step, we then take "adaptive_Flight_controller" which we then proceed to the second row and ask the same questions again:

Table **Error! No text of specified style in document.**.33 Example 4WnH processes applied to the Eagle Robot case study

Source situation	**interaction**	**Sink situation (What to influence)**	**How to control it? (control)**	**What to appreciate? (appreciation)**
Eagle_drone	+flying	low_patrolling _altitude	+Regulate **adaptive_ Flight_controller**	+Visually scanning visible_terrain, GPS
Adaptive_Flight _controller	+Regulate	Variable_&_fix_ altitude	+Adjust continuous_ rotor_speed	+Monitor precise_barometric _sensor, GPS

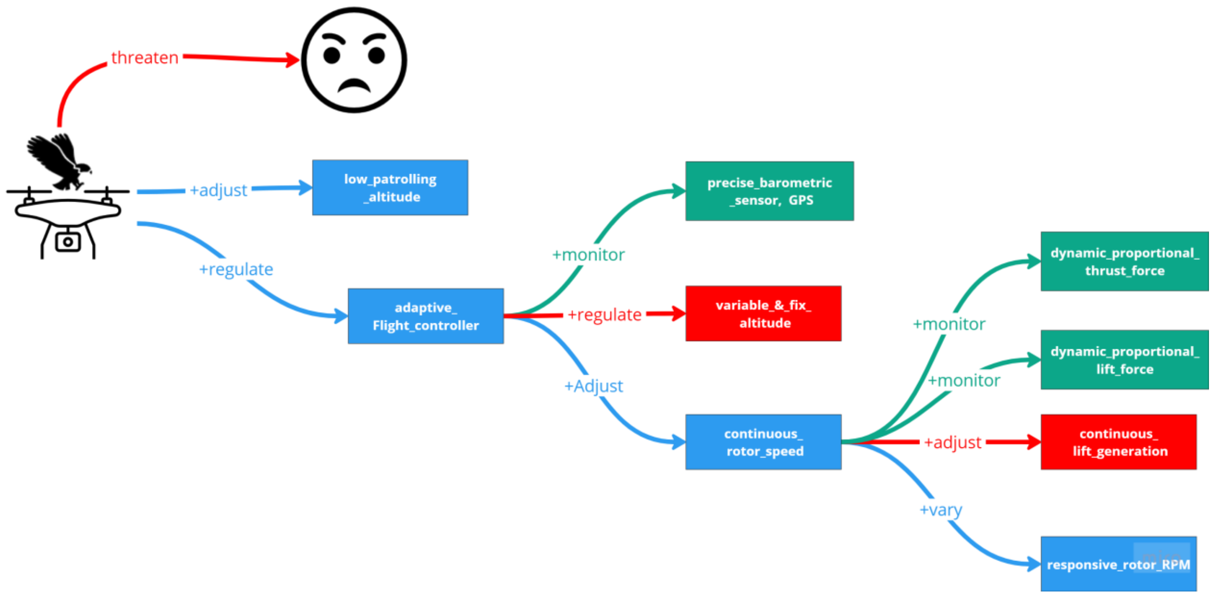


Figure **Error! No text of specified style in document.**.16 AIC hierarchical modelling schema for 4WnH analysis example taken from H.6.4.1.d.

F.10 Developing a CuneiForm-based Training Syllabus

The input to this stage would be "ML Development Dataset Requirement", which is typically derived from a training concept that mitigates the safe operating concept (safety requirements). The idea is to conceptually reinterpret the safety requirements through the training concept and datasets to abstract CuneiForms, which are instantiated through datasets. The validation of the CuneiForms provides evidence that datasets effectively mitigate safety concerns. We use the term training here as a general term, not specifically meaning a training dataset only. Training, validation and testing datasets are refinements of a training syllabus. We view the ML model like a trainee apprentice, and the architect like a trainer. Training syllabus is analogous to training curriculum, where a trainee model is set to be taught about different aspects of the complexity of the human world. The trainer (architect) needs to communicate the physical world by teaching pictorial cards to the machine. These teaching pictorial cards are the CuneiForms, which constitute the Training syllabus.

F.10.1 Step A) Articulate the pictorial problem context:

The first step towards characterising a CuneiForm for a dataset requires a detailed articulation of the types of objects, dynamics, and relationships among objects. In this step, we look at the environment from TOI's perspective (problem perceptive). It is possible to derive multiple problem situation contexts for every requirement. It is also possible to derive multiple

CuneiForms for every context in pictorial situations. Think of the pictorial situation as a complex of complexes problems to which the perception observes and applies solutions. For such a task, we will adapt the AIC-based Chain-of-Thought problem articulation technique [5] to predict objects and behaviours for CuneiForm characterisation.

In our systems approach, we defined that for any complex to achieve influence purpose, it must control some other complexes. The complex must appreciate other environmental complexes to guarantee that the control behaviour achieves the intended influence. In this application, we readapt the latter intuition by considering what complexes a TOI aims to influence. Then, imagine what complexes a TOI must appreciate to achieve influence. Then, imagine what related aspects to the appreciated complexes a TOI should exploit or govern (control). The output of every step of the Chain-of-Thought is a prediction made by the architect about a potential scenario. In our case, it will be a prediction about a situation that the perception system may encounter, and every step will add an aspect of some pictorial situation, which will then be captured in multiple CuneiForms.

In every step, consider the possibility of the perception system encountering various complexity levels, ranging from best-case to worst-case scenarios:

Step 1) Define the TOIs and their pictorial appearances:

Considering the given requirement, we can obviously notice that a TOI would be an adversarial drone. Also, the requirement demands a plurality of drones with unusual appearances. Given these ideas, we can imagine a visually problematic situation of having more than one drone flying, which employs some stealth technique. One possible stealth technique is camouflage (for a different CuneiForm, we may choose a different stealth technique). This step will help define TOI's characteristics.

Hence, **architect prediction:** the architect asserts that the perception system may face a pictorial situation with two flying adversarial drones in no precise formation. Neither of these is a typical drone design and has a different camouflaged skin.

Step 2) Consider other objects that TOIs aim to influence:

Given the output from Step 1 and the conditions required for the environment to be in the train track zone, we can imagine that adversarial drones aim to influence the open airspace of the train track zone. This step will help define the background object characteristics for the CuneiForm.

Hence, **architect prediction:** the architect asserts that adversarial drones aim to influence the safety and security of open spaces in the train track zone.

Step 3) Consider objects that TOIs must appreciate:

Consider what other objects the TOIs cannot influence or control within the immediate operational environment, which TOIs must appreciate for their behaviour to guarantee the successful achievement of their influence purpose. Reflect the output from step 2 and the terrain of the operational environment (potential environmental scenery). This step predicts what other external complexes (to the TOI) within or outside the problem environment must be appreciated. For example, the adversarial drones must appreciate the following potential environmental objects and behaviours in order to achieve successful stealth and influence track zone open space: The time of the day because of the effect of visibility, the density of train traffic because they dictate the time of the day that is best for the adversarial drone to be around, trees and their density because they affect the usefulness of the camouflage, the usefulness of the camouflage. Since we consider trees, we may also consider local birds that may reveal adversarial drone cover.

Local people in the surrounding houses can blow up adversarial drone cover and security cameras. These objects demand that adversarial drones behave differently, which will help derive more background objects' characteristics. Hence, **architect prediction:** the architect asserts that the adversarial drones may have to appreciate the following environmental scenery aspects: the time of the day, trains, trees, local birds, local people and security cameras.

Step 4) Consider what other objects TOIs must control:

Consider what other objects TOIs must govern, exploit, or make use of to achieve their intended influence. Considering the output from step 3, we can imagine that adversarial drones must make use of the area to influence the safety and security of open space while maintaining stealth. It must intelligently govern and utilise the following aspects: tree area density (which parts of the trees are best suited for hiding), finding the optimum path around houses to avoid sighting by local people. The sun's location in the sky may affect its perception system. The blind spot of existing security cameras. Hence, **architect prediction:** architect asserts that adversarial drones autonomously look to exploit the following aspects: trees' dense vegetation, security cameras' blind spots, and areas among the terrain where local people are less attentive and may avoid local bird nests.

Step 5) Produce pictorial problem context:

Collate all the knowledge predicted from the above Chain of Thought and articulate a pictorial situation: Gather all predictions from the above steps and annotate aspects that will be represented in a pictorial sense (in abstract icons).

Pictorial problem context: The architect asserts that the perception system may face a pictorial situation with two flying adversarial drones in no precise formation {1}. These are not typical drone designs {2} and have different camouflaged skin {3}. Adversarial drones aim to influence the safety and security of open space {12} in the train track zone{4}. In order to achieve the purpose of influence, the adversarial drone may have to appreciate the following environmental scenery aspects: the time of the day{5}, trains{6}, trees{7}, local birds{8}, local people{9}, local houses{10}, and security cameras{11}. To achieve an appreciation of appreciated complexes, adversarial drones autonomously look to exploit the following aspects: trees' dense vegetation{F.1}, security cameras' blind spots{11.1}, and areas among the terrain where local people are less attentive{9.1} and may avoid local bird nests{8.1}.

Table 5.16 captures the summary of steps:

Table **Error! No text of specified style in document.**.34 CuneiForm Pictorial situation articulation

Pictorial Situation CoT step	Definition
Step 1) Define the TOIs and their pictorial appearances	**Architect prediction: The** architect asserts that the perception system may face a pictorial situation with two flying adversarial drones in no precise formation. Neither of these is a typical drone design and has a different camouflaged skin.
Step 2) Consider other objects that TOIs aim to influence	**Architect prediction:** The architect asserts that adversarial drones aim to influence the safety and security of open spaces in the train track zone.
Step 3) Consider objects that TOIs must appreciate	**architect prediction:** the architect asserts that the adversarial drones may have to appreciate the following environmental scenery aspects: the time of the day, trains, trees, local birds, local people and security cameras.
Step 4) Consider what other objects TOIs must control (target)	**architect prediction:** architect asserts that adversarial drones autonomously look to exploit the following aspects: trees' dense vegetation, security cameras' blind spots, and areas among the terrain where local people are less attentive and may avoid local bird nests.
Step 5) Produce pictorial problem context	**Pictorial problem context:** The architect asserts that the perception system may face a pictorial situation with two flying adversarial drones in no precise formation {1}. These are not typical drone designs {2} and have

	different camouflaged skin {3}. Adversarial drones aim to influence the safety and security of open space {12} in the train track zone{4}. In order to achieve the purpose of influence, the adversarial drone may have to appreciate the following environmental scenery aspects: the time of the day{5}, trains{6}, trees{7}, local birds{8}, local people{9}, local houses{10}, and security cameras{11}. To achieve an appreciation of appreciated complexes, adversarial drones autonomously look to exploit the following aspects: trees' dense vegetation{F.1}, security cameras' blind spots{11.1}, and areas among the terrain where local people are less attentive{9.1} and may avoid local bird nests{8.1}.

F.10.2 Step B) Characterise the CuneiForm Training Classes:

In Step A, after generating the pictorial situation context, we define a CuneiForm characteristic that captures a worldview of the pictorial situation, a training class part of a Training syllabus for which the Eagle Drone must be trained. The architect may identify several CuneiForm abstraction images for every situation predicted from process A. The architect may refine the output step A in more detail in this step by defining CuneiForm training class that will design the experience in the pictorial situation. Each characteristic is represented and captured in an icon, which captures one descriptive idea. Step A looked at the environment from the TOI perspective; step B was about looking at the whole environment with a TOI. Table F.24 represents each characteristic training class and its associated depictive definition that captures one possible situation. "**Training Class Primary Purpose"** is not part of the implementation table but instead describes the purpose of the class.

Table **Error! No text of specified style in document.**.35 Characteristic Training Classes definitions for a CuneiForm abstract image

CuneiForm Characteristic Training Class	**Training Class Primary Purpose [not part of the implementation table]**	**Definition [Example]**

Step 1) Define visible horizon attitude.	The primary purpose of this training class is to train the model to appreciate the visible horizon of the real world.	Level horizon: Frame roll = 0°, Frame pitch = 0° {13}
Step 2) Define all the TOIs and their aesthetic complexity. Then, generate abstract representative icons for the CuneiForm abstract image.	The primary purpose of this training class is to train the model to appreciate the characteristics of Targets of Interest (TOIs).	For example, two adversarial drones{1}, tri-copters {2}, are camouflaged in different green patterns {3}.
Step 3) Define TOI's motion trajectory and dynamic optical situations. Then, update the generated abstract representative icons for the CuneiForm abstract image.	The primary purpose of this training class is to train the model to appreciate the dynamic characteristics of the TOIs.	Motion trajectory: Linear motion captured in consecutive images where the drone appears to move in a straight line at a constant speed (no acceleration) {1.1}. Dynamic optical state: captured without optical blur {1.2}.
Step 4) Define the background objects associated with TOIs and environmental scenery in the background of the CuneiForm. Then, generate abstract representative icons for the CuneiForm abstract image.	The primary purpose of this training class is to train the model to appreciate the characteristics of the background objects alongside the TOIs.	For example, Houses are to be depicted in a typical British house {10}, with red brick masonry designs {10.1}, and train tracks in any form {4}. The sky as open space {12}. The ground-level separation from the sky as a line dividing the image {12.1}
Step 5) Define the background Objects' Motion situations and dynamic optical situations. Then, update	The primary purpose of this training class is to train the model to appreciate the dynamic characteristics of	Background objects' motion trajectory is static {4.1, 10.2} Dynamic optical state: no motion blur {4.2, 10.3}

the generated abstract representative icons for the CuneiForm abstract image.	the background objects alongside the TOIs.	
Step 6) Define TOI's positioning in the CuneiForm. Then, generate abstract representative icons for the CuneiForm abstract image.	The primary purpose of this training class is to train the model to appreciate the characteristics of the TOIs localisation behaviour.	Using the general definition of positioning compartments in Figure 4, we define drone 1 to be in the up_center position {2.1} and drone 2 to be in the centre position {2.2}.
Step 7) Define TOI's 3D orientation. Then, update the generated abstract representative icons for the CuneiForm abstract image.	The primary purpose of this training class is to train the model to appreciate the characteristics of the TOI's 3D orientational variations.	Using the general orientation definition in Figure 3, we define drone 1 to be in front_up_right orientation {2.3} and drone 2 to be in front_down_left orientation {2.4}.
Step 8) Define the optical distance for each TOI in nindans. Then, update the generated abstract representative icons for the CuneiForm abstract image.	The primary purpose of this training class is to train the model to appreciate the characteristics of the TOIs visual scale.	Both drones are represented in a pictorial distance of 1 nindan (equivalent to 1/9 total area of pictorial frame) {2.5}
Step 9) Design the relevant icons to produce the CuneiForm and give an example of an instantiating image	The output of this process would be a CuneiForm abstract image, shown in Figure 5.17, and an example instantiating image, shown in Figure 5.18.	

Figure 5.17 presents a CuneiForm abstract image that characterises a dataset, integrating the defined characteristics from Table 2. The dashed grid showcases the 9 compartments. The traceability annotations next to each icon link to Table 4.17, which connects to the articulated pictorial situation and the safety requirement example. The CuneiForm is then used as a guideline for ML engineers to gather images demonstrating the CuneiForm. Every CuneiForm should include an ideal example to assist with finding instantiating images. Figure 5.18 depicts a

compliant instantiating image for the produced CuneiForm. Conceptually, Figure 5.18 meets the attributes of the parent CuneiForm.

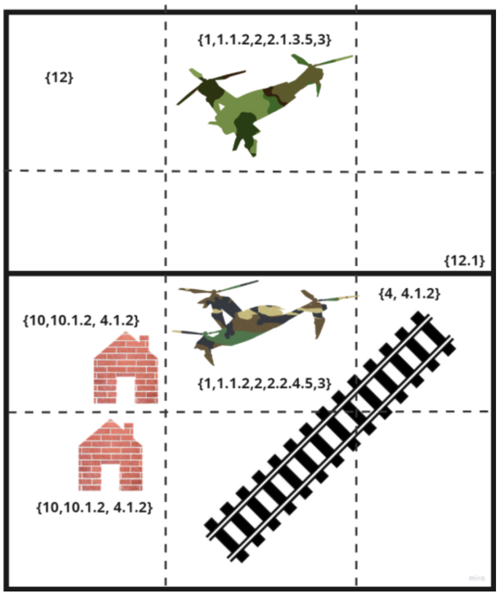


Figure **Error! No text of specified style in document.**.17 A CuneiForm abstract image

Our process does not look for a precise match but a conceptually accurate mapping between the concrete image (Figure 5.18) and its parent abstract CuneiForm. The way to derive Figure 5.18 is by placing two drones (tri-copters) in the same places as stated in Figure 5.17, then placing red brick houses somewhere in the background where the partition of the horizon is distinguished. Then, ensure that the train tracks are cutting across the image.

If the main features are included in image instances, extra features such as trees or clouds may emerge to be included (due to operational environment complexity). Extra features outside the CuneiForm definition must be flagged and justified in the associated documentation. The Architect may design another CuneiForm to cover those extra features or edit the existing CuneiForm with the extra emergent features and trace them to the requirement.

Figure **Error! No text of specified style in document.**.18 Example instantiated image of the designed CuneiForm.

F.10.3 CuneiForm Design Artefact

The CuneiForm Design Artefact process provides a structured process for defining and modelling the requirements, contexts, and configurations necessary to train and validate perception systems, such as those used in Eagle Drone computer vision models. It emphasises the generation of robust, generalised datasets that enable the system to identify and respond to adversarial drones under a wide range of visual and operational conditions. The process bridges high-level design requirements with detailed pictorial scenarios to ensure comprehensive coverage of complicated environments.

(1) A CuneiForm Training Syllabus Characteristics

The following is a list of training classes:

i. **Visible Horizon Attitude:** It specifies the orientation of the visual frame to maintain consistency across training images. A level horizon with a frame roll and pitch of 0° ensures uniformity in visual perspective, reducing noise caused by varying angles and facilitating robust feature extraction.

ii. **TOIs Definition and Aesthetic Complexity**: This section describes the characteristics of TOIs, central to the training objectives. The TOIs include specific entities, such as adversarial drones, defined by their structural type (e.g., tri-copters) and visual complexity (e.g., green camouflage patterns). These variations simulate diverse operational scenarios, enhancing the model's generalisation ability across different appearances and contexts.

iii. **TOI Motion and Dynamic Optical Situations**: This section details the motion trajectories and optical situations of TOIs. Linear motion with constant speed ensures predictable dynamics, allowing the model to learn consistent patterns. Dynamic optical situations are designed to avoid artefacts such as motion blur, ensuring clear feature representation and reducing training noise.

iv. **Background Objects Associated with TOIs:** Specify the environmental context in which TOIs appear. Background objects include houses with typical architectural features (e.g., red brick masonry) and open-space skies. These contextual details ground the TOIs in realistic scenarios, supporting developing a model sensitive to environmental variations.

v. **Background Objects Motion and Dynamic Optical Situations:** Ensures static trajectories for background objects to prevent distractions and unintended feature correlations during training. Optical situations are designed to be free of motion blur, maintaining visual clarity and aiding accurate model focus on relevant details.

vi. **TOI's Pictorial Positioning and Distance:** Defines the spatial arrangement of TOIs within the image frame to simulate realistic operational perspectives. Specific positions, such as "up centre" or "centre," and distances (e.g., one nindan, equivalent to 1/9 of the frame area) provide diversity in positioning, ensuring robust feature detection across various scenarios.

vii. **TOI's 3D Orientation:** This specifies the 3D orientation of TOIs (e.g., front upright, front down left) to introduce variability in perspectives. These variations simulate different observation angles, enhancing the model's capability to recognise targets under diverse orientations.

(2) Implementation Steps

The overall process of how to derive a training syllabus can be summarised in the following:

i. **Perception Safety Training Requirement**: Define the requirements for the perception system, such as identifying stealthy or unusual adversarial drones, while ensuring robustness across environmental and situational variabilities.

ii. **Pictorial problem Context**: Establish the operational and environmental context in which adversarial drones interact with other elements (e.g., trees, people, cameras) to influence security. This step considers dynamic factors like blind spots or camouflage strategies.

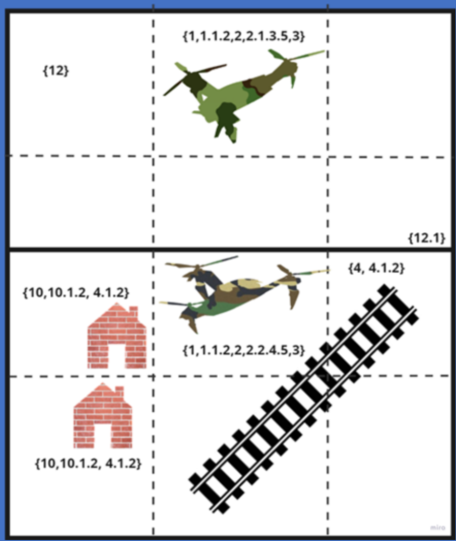
iii. **CuneiForm Training syllabus Characteristics**: Translate the situational context into training class specifications, detailing the appearance, motion, and spatial arrangements of targets and background objects.

iv. **Instantiated Concrete Image Example**: Generate representative images based on the abstract characteristics, ensuring alignment with training objectives and system requirements.

To capture the CuneiForm design outcome, we propose that the following artefact (Table 5.18) can be included as evidence for the AMLAS Safety Case (Artefact [L]: Data Requirements).

Table **Error! No text of specified style in document.**.36 Example of a Cuneiform Training syllabus

ML Development Dataset Requirement	Eagle Drone ML component shall be trained to enable a robust and generalised computer vision model to identify various adversarial drones, especially unusual or stealthy ones.
Pictorial problem context	the architect asserts that the perception system may face a pictorial situation with two flying adversarial drones in no precise formation {1}. These are not typical drone designs {2} and have different camouflaged skin {3}. Adversarial drones aim to

	influence the safety and security of open space {12} in the train track zone{4}. In order to achieve the purpose of influence, the adversarial drone may have to appreciate the following environmental scenery aspects: the time of the day{5}, trains{6}, trees{7}, local birds{8}, local people{9}, local houses{10}, and security cameras{11}. To achieve an appreciation of appreciated complexes, adversarial drones autonomously look to exploit the following aspects: trees' dense vegetation{7 .1}, security cameras' blind spots{11.1}, and areas among the terrain where local people are less attentive{9.1} and may avoid local bird nests{8.1}.	
CuneiForm Training syllabus Design		
Abstract CuneiForm Characteristics (Training Classes)	**Abstract CuneiForm Training Classes Characteristics definitions**	**Output CuneiForm and an example image**
Visible horizon attitude	Level horizon: Frame roll = 0°, Frame pitch = 0°{12.1}	CuneiForm Abstract Image: 
TOIs definition and their aesthetic complexity	Two adversarial drones{1}, tri-copters {2}, are camouflaged in different green patterns {3}.	
TOI Motion and Dynamic optical situations	Motion trajectory: Linear motion captured in consecutive images where the drone appears to move in a straight line at a constant speed (no acceleration) {1.1}. Dynamic optical state: captured without optical blur {1.2}.	
Background Objects associated with TOIs	Houses are to be depicted in a typical British house {10}, with red brick masonry designs {10.1}, and train tracks in any form {4}. The sky as open space {12}.	Instantiated Concrete Image example:
Background Objects Motion and Dynamic	Background objects' motion trajectory is static {4.1, 10.2}	

optical situations	Dynamic optical state: no motion blur {4.2, 10.3}			
TOI's Pictorial Positioning		Up Centre {2.1}		
		Centre {2.2}		
TOI's 3D Orientation			Front up right {2.3}	
	Front down left {2.4}			
TOI's Pictorial Distance	Both drones are represented in a pictorial distance of 1 nindan (equivalent to 1/9 total area of pictorial frame) {2.5}			

The structure of the CuneiForm artefact is merely a guideline. The architect may alter the artefact as needed for the process; for instance, in our AVOIDDS case study, we required a different approach to represent the CuneiForm to suit the application since we were designing the CuneiForm retrospectively.

F.11 CuneiForm Validation Process

The validation process is encouraged to rely on as much manual effort as practically possible. It is feasible to employ some automated techniques, provided we use deterministic means, to assist with efficiently validating large datasets. However, the architect should refrain from using AI or non-deterministic methods to perform the validation. The reason is the dilemma of "who validates the AI validator?". Human intelligence should be considered as the final authority over machine intelligence behaviour. Furthermore, it is essential to note that some degree of imprecision is allowed when validating the CuneiForms since this is not a mathematically precise activity but rather a rationally accurate operation.

In other words, we aim or hope for "human-driven rational accuracy", not "mathematical, logical precision" between the validated images and the parent CuneiForms. However, we attempt to reduce the uncertainty in the validation output by using manual or automated techniques.

Mathematical precision is welcomed where it is practical. For example, validating the pictorial distance of TOI can be done formally by computing the ratio of occupied pixels by the TOI and the number of total pixels of an image. While validating the attitude of the pictorial horizon is done using visual tools that are not exact but rather accurate enough.

The following are some manual techniques for performing the validation. We developed these techniques to assist us in reverse engineering the CuneiForms, which would have been associated with the AVOIDDS training datasets based on the characteristics of the training sample. We then use them to validate the dataset in the AVOIDDS case study and produce the validation report (see Appendix I, section I.10). Human involvement in the design validation is crucial.

F.11.1 Vis&V: Visualise and Validate

This technique presents validators with both CuneiForm definitions and corresponding dataset images. Validators visually inspect and judge if the image set reflects CuneiForm's descriptive intent. This process embraces interpretative flexibility; perfect agreement among individuals isn't necessary, but consensus among a multidisciplinary team (including systems thinkers) provides sufficient validation. Judgement outcomes are documented in a validation report (e.g. as in AVOIDDS, Appendix I.10) to support safety case evidence.

F.11.2 Validating Visual Dimensions 1–3

Each CuneiForm specifies certain *visual dimensions* that must be evaluated:

- **Visual Dimension 1: Visual Aesthetics –** Validate the overall visual style or appearance concerning the CuneiForm reference.
- **Visual Dimension 2: Perceived Motion & Optical Effects –** Determine if the image conveys the intended motion characteristics or visual illusions (e.g., blurring).
- **Visual Dimension 3: 3D Orientation –** Assess whether the object's (TOI's) spatial pose matches the expected three-dimensional positioning.

Each dimension is assessed through Vis&V, supported by engineering rationale or justification for each verdict.

F.11.3 Validating Visual Dimension 4: Pictorial Positioning Zones

This dimension focuses on spatial placement within the image frame. Validation involves:

1. Overlaying an opaque mask divided into zones (e.g. quadrants).
2. Locating the TOI (Target of Interest) within the image.
3. Confirm that its position aligns with the zone defined by the CuneiForm.

This step may be script-assisted for efficiency but requires human cross-verification to ensure conformance.

F.11.4 Validating Visual Dimension 4: TOI's pictorial positioning zones definition

To validate the necessary postponement of the CuneiForm, the human validator can use the CuneiForm Canvas, laying it out on top of the validated image. This way, it can be confirmed that the instantiating image captures the CuneiForm. Figure 5.19 demonstrates the activity:

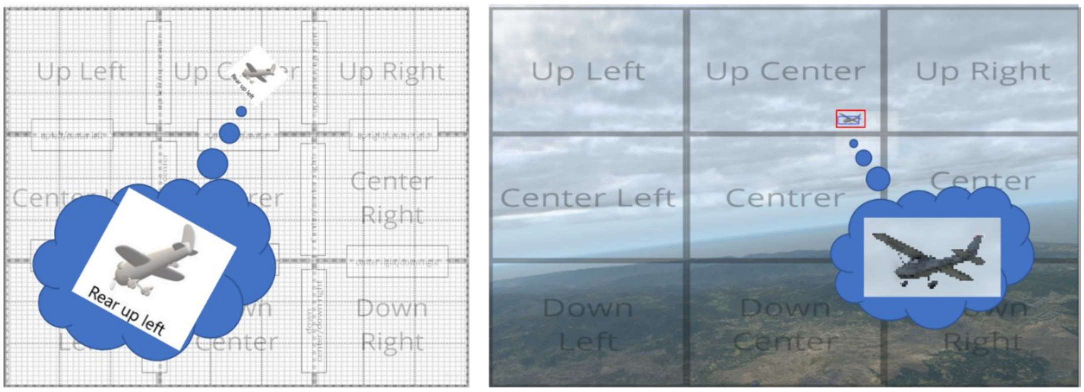


Figure **Error! No text of specified style in document.**.19 Example validation of TOI positioning.

The image on the right represents the CuneiForm positioning canvas. The picture on the left represents an example of an instantiating image for some cuneiform. We can see that the TOI is positioned in the Up Center quadrant, thus validating the CuneiForm on the right.

F.11.5 Validating Visual Dimension 5: TOI's visible pictorial distance

To validate the pictorial distance of a TOI in an image for a CuneiForm, we can follow an automatic process using a Python script that extracts the number of pixels occupied by a labelled image bounding box and calculates the ratio to the size of the image itself. See section 4.5.5.5 for more details. Alternatively, the validator could utilise the CuneiForm canvas and count the number of squares occupied by the bounding box manually.

F.11.6 Validating background objects and their dynamic situations

Like TOIs and their dynamic situations, the human validator can perform a sanity check to confirm that the background objects in the validated image capture the cuneiform characteristics.

F.11.7 Validating Pictorial Representation of Visible Horizon Attitude

Although some aspects of CuneiForm can be automated, such as pictorial distance, validating whether an image truly captures the CuneiForm requires another machine learning model to find the horizon and measure its attitude. As mentioned previously, to validate the dataset, it is best to refrain from entirely relying on a machine learning validation system, as the question arises: who validates the validator? A human manual validation is essential in this case, where a human validator performs the following process. We will need to use the PHI stencil (Figure 5.20):

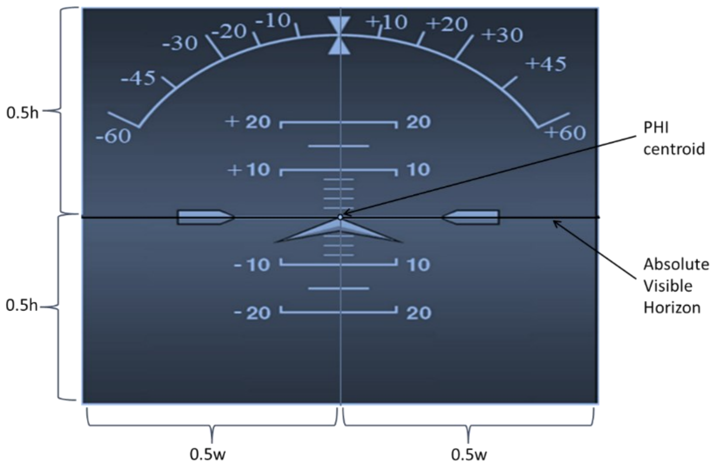
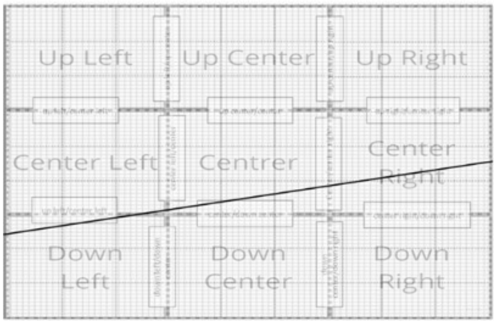


Figure **Error! No text of specified style in document.**.20 Pictorial Visible Horizon Attitude Indicator (PHI) stencil

Let’s assume the following CuneiForm horizon characteristics (Table 5.19):

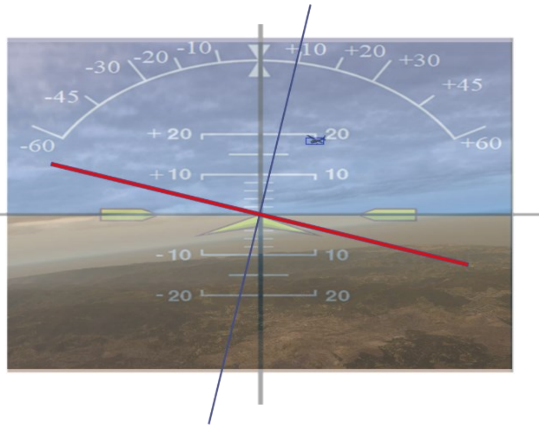
Table **Error! No text of specified style in document.**.37 Example CuneiForm characterisation of pictorial horizon

Horizon attitude category	**Roll**	**Pitch**
Negatively Tilted Lowered Horizon $-90 \le ROLL < -1$, $-45 \le PITCH < -1$	-5	-5

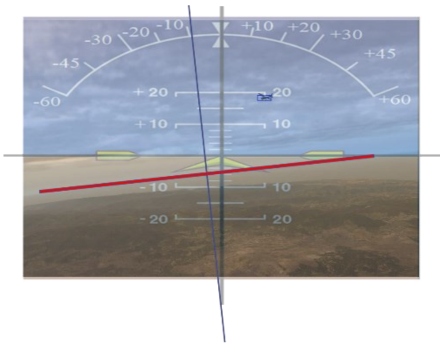


To validate that an instantiated image captures the CuneiForm, we perform the following manual activities:

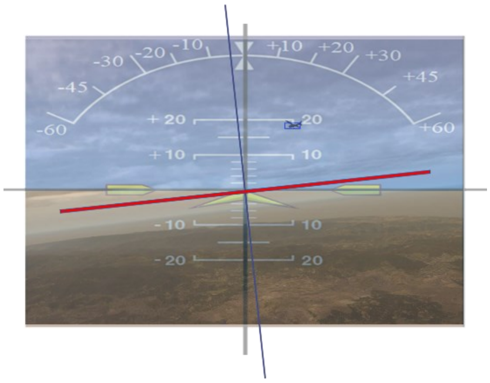
Table **Error! No text of specified style in document.**.38 Pictorial horizon validation process

<table>
<tr><th>Validation step</th><th>Results</th></tr>
<tr><td>Step 1) Pick the required image for validation.</td><td></td></tr>
<tr><td>Step 2) Place the PHI stencil and resize it to cover the image like the overlay in previous figures. This ensures that the stencil fully aligns with the image boundaries and that the perceived horizon falls within measurable areas. Also, place the Horizon Indication Measurement Ruler (HIMR).</td><td>
</td></tr>
<tr><td>Step 3) Validate the roll angle:

3a: Rotate the HIMR and precisely align the red horizontal like of the HIMR to match exactly where you think the horizon is.

3b: Then drag the HIMR, maintaining the angle of tilt, and precisely align the central point on top of each other. Then, measure where the vertical blue line of the HIMR indicates to.</td><td>Step 3a


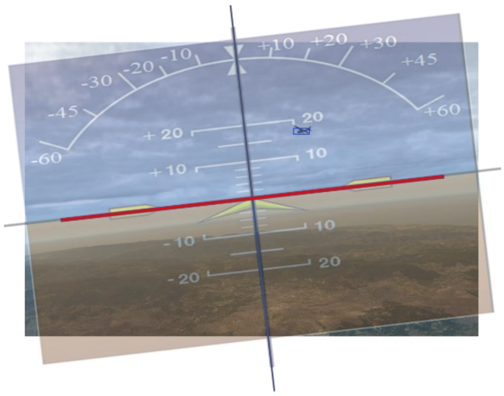
Step 3b</td></tr>
</table>

<table>
<tr><td></td><td>

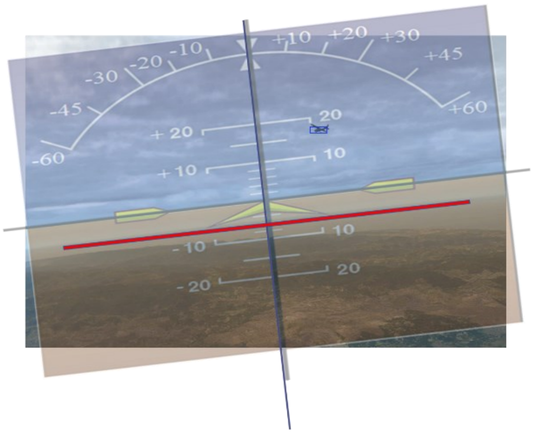
Roll angle = -5</td></tr>
<tr><td>Step 4) Validate the pitch angle:

4a: To validate the pitch angle, rotate the entire PHI image so the blue line makes the red line level (0 degrees roll). This aligns the image frame horizontally and neutralises the roll, making pitch estimation more intuitive.

4b: Then, lower the HIMR until the red line covers the image horizon.</td><td>Step 4a



Step 4b



Pitch angle = -5</td></tr>
<tr><td colspan="2">Step 5) Repeat this process for a representative sample of images. This helps validate the dataset's diversity and correctness of horizon attitudes, particularly in complex scenarios (e.g., mountainous terrain, urban landscapes, or maritime scenes with fluid horizons).</td></tr>
</table>

Step 6) Document any inconsistencies or unusual patterns. If specific frames show inconsistent horizon attitudes (e.g., a tilted horizon in a supposedly level horizon in the CuneiForm), such samples indicate that the original population of images may contain inconsistent instantiations that designate CuneiForm. It may require reannotation or exclusion from training datasets.

Table 5.20 describes the validation process for the horizon roll and pitch angles using PHI. Appendix D captures our CuneiForm validation report for the AVOIDDS case study. Section 7.8 gives an example of how we validated the AVOID dataset.