Appendix I Mid-air Collision Avoidance Case Study

In this Appendix, we explore the application of the AIC (Application, Influence, and Control) Systems Approach to enhance AVOIDDS (Aircraft Vision-Based Object Identification and Detection Dataset System), a dataset created to advance vision-based aircraft detection capabilities. This approach, grounded in systems thinking, will allow us to better structure and address the complexities inherent in detecting intruding aircraft, especially those relevant to autonomous aircraft systems designed to perform collision avoidance manoeuvres without human input. Through AIC, we aim to delineate the scope of the detection problem, address the impact of various influencing factors on system performance, and control the dataset's development to ensure it accurately reflects real-world scenarios.

I.1 Abstract

See Appendix M

I.2 Introduction

See Appendix M

I.3 Thought Experiment

See Appendix M

I.4 Stage 1: Uncertainty Problem Articulation and Operational Environment Modelling[1]

Problem brief:

> Two or more aircraft come into unplanned contact during flight, leading to a possible mid-air collision, posing a significant safety threat. These incidents often result in severe damage or destruction due to high velocities and potential subsequent impacts. The by-passing aircraft is in constant motion, following a flight path that intersects with another aircraft.

I.4.1 Predictive Thinking Pipeline 1: Appreciate the Complex of the Problem Complexity Field

At this stage of the thinking process, the architect is directed to utilise the key concept of E.2.9: System Primary Purpose (PrimeP). To identify the primary purpose, the following considerations are necessary:

- Recognising the relevant systems within the problem.
- Identifying the Supra Complexes related to the problematic systems.
- Defining the primary purpose of these Supra Complexes.

Additionally, we are examining the concepts of Confusing Complex and systems. The confusing aspect of the problem description indicates an unclear purpose or an undesirable, conflicting emergent purpose that contradicts the architect's intent. The definition of appreciation limits the appreciative system's effect on the observed phenomenon, which poses a challenge for us to address. Our initial aim is to transform the observed complexity dynamics to enhance the train track zone's capacity for impact situations.

I.4.1.1 Step 1.1) Identify a list of unsafe or confusing behaviours

In this step, we applied General Systems Rules to identify unsafe or confusing behaviours within the context of mid-air collisions. This process began with examining the problem of aircraft coming into unplanned contact due to deviations or intrusions in controlled airspace. We then assessed behaviours that disrupt the intended harmony of safe air travel, particularly those that introduce ambiguity or risk to pilots and air traffic control (ATC).

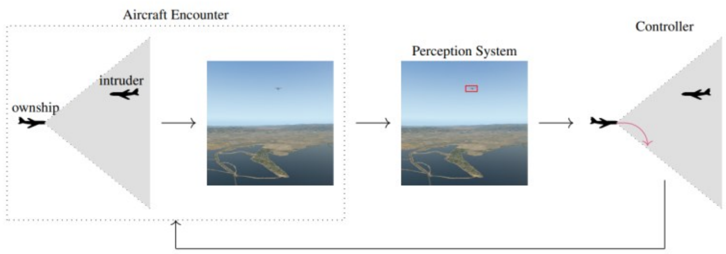
---

[1] For a summary, see Section M.2

Using the rules, we identified confusing behaviours like unanticipated flight path deviations and erratic movements by unauthorised aircraft, which do not align with the intended purpose of maintaining flight separation. Each behaviour was analysed for its unsafe aspects and potential undesirable outcomes, such as increased collision risks or reactive manoeuvres by other aircraft. This analysis highlighted complexities in ATC's task to ensure safety and underscored the need for real-time tracking and adaptive response strategies.

This step's outputs focused on describing problem situations that increase collision risk, without introducing solutions, ensuring a thorough depiction of a "Confusing Complex" within aviation safety.

Table I.1 Identification and analysis of unsafe problematic behaviours in mid-air collision

SECoT Title	Unsafe problematic behaviours list identification
Input	In aviation, mid-air collisions, where two or more aircraft come into unplanned contact during flight, pose a significant safety threat. These incidents, often resulting in severe damage or destruction due to high velocities and potential subsequent impacts. The by-passing aircraft is in constant motion, following a flight path that intersects with another aircraft.  Diagram taken from [1]
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 Method	**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 criteria1.1.3:** The step is regarded as complete when a problem situation is described, including undesirable impact.
Output Prediction	**Architect assertion 1.1.4:** **Confusing and Unsafe Behaviours Identified:** 1. **Aircraft Deviating Unexpectedly from Assigned Flight Path:** o **Description:** An aircraft deviates from its assigned path, intersecting with a nearby flight path in the process. o **Unsafe Aspect:** This behaviour contradicts expected adherence to assigned routes, introducing ambiguity about the aircraft's intentions and position for other pilots and ATC.

	◦ **Undesirable Outcome:** This unplanned deviation could increase collision risk as nearby aircraft or ATC might not anticipate this course change, leading to unsafe proximity or even a direct collision. 2. **By-passing Aircraft Moving Erratically at a High Altitude near Crowded Airspace:** ◦ **Description:** An unauthorised or unplanned aircraft enters high-altitude airspace without coordinated communication, moving in a way that suggests erratic or unpredictable flight patterns. ◦ **Unsafe Aspect:** Erratic movement within controlled airspace is unexpected and creates difficulty for ATC in predicting the bypassing's path, especially if the aircraft does not respond to communication. ◦ **Undesirable Outcome:** This unpredictability can obstruct ATC's ability to route nearby aircraft safely and may cause surrounding aircraft to take evasive actions, increasing the likelihood of incidental close calls or collision.

In Table I.1, a quality or design assurance authority can see why the design excludes certain aspects. This would allow us to scrutinise the design thinking right from the initial stages, thus enabling more transparent thought processes that lead to decisions. At this stage, soft aspects such as ethics can be reviewed objectively and ascertained as part of evidence collection to support an informative safety case.

The output is acceptable if it captures the qualities of a Confusing Complex definition. The output is considered unacceptable if the description fulfils the following criterion in unacceptable production:

- The behaviour involves solutions and not problems.
- The involved parts do not fall within each other's sphere of concern or the observing architect's sphere of concern.
- The emergent outcomes are desirable.
- The interacting coexisting parts seem to be working as a clear system.

If the output description satisfies the above notions, the problem is not a Confusing Complex.

Implementation: The behaviours observed in this scenario reflect that aircraft and ATC are not operating in the expected way, creating confusing complexities in air traffic control and flight safety. Mid-air collisions become more likely when aircraft deviate unexpectedly from expected paths or fail to communicate. This complexity requires robust real-time tracking and adaptive ATC strategies to handle deviations that may otherwise lead to unsafe proximity or collisions.

I.4.1.2 Step 1.2) Generate a descriptive image that visualises the unsafe behaviour

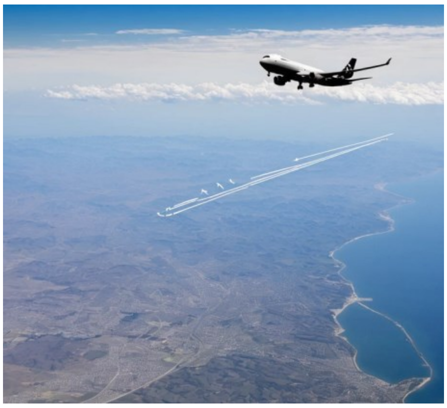
In this step, we focused on generating a descriptive image to visualise the unsafe behaviour of an aircraft deviating unexpectedly from its assigned flight path. This step aimed to complement the previously articulated human-written description with a visual representation to enhance lateral thinking and facilitate a clearer understanding of the situation. We began with the architect's assertion regarding the ambiguity and risks associated with an aircraft's deviation, which could increase the likelihood of collisions due to the unpredictability of its trajectory. Using the general systems rules, we identified the key elements of this confusing behaviour and explored how they could be graphically depicted.

We crafted a prompt for the DALL-E tool to generate a simplistic sketch illustrating the scenario, which included the deviating aircraft, nearby aircraft on expected paths, and background elements like clouds. The output was evaluated based on whether it captured the essential characteristics of the confusing behaviour and accurately depicted the interactions among the elements involved. This visual representation not only encapsulated the architect's perspective on the problem but also served as a basis for discussions among the design authority team and stakeholders, allowing them to challenge or validate the assumptions made about the situation.

Table I.2 Visual representation of unsafe and confusing drone behaviours in train track zones

SECoT Title	Unsafe problematic behaviours visualisation
Input	**Architect assertion 1.1.4:** **Aircraft Deviating Unexpectedly from Assigned Flight Path:** ○ **Description:** An aircraft deviates from its assigned path, intersecting with a nearby flight path in the process. ○ **Unsafe Aspect:** This behaviour contradicts expected adherence to assigned routes, introducing ambiguity about the aircraft’s intentions and position for other pilots and ATC.

	○ **Undesirable Outcome:** This unplanned deviation could increase collision risk as nearby aircraft or ATC might not anticipate this course change, leading to unsafe proximity or even a direct collision.
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 Method	**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:** The architect asserts that the following depiction model captures the confusing behaviour faithfully:



Note that the visualisation clearly describes how the architect perceived the situation. This would allow the design authority team to question the accuracy of the architect's assessment of the problem. Furthermore, stakeholders may argue for or against the assumptions presented.

A depiction does not comply with the step if it misses key elements or depicts an unrealistic (not physically possible for it to be present on the ground within the physical boundaries of the problem domain) visualisation of the problem.

Implementation: Use the Dall-E tool and the following prompt:

"Generate a very Simplistic Sketch, that depicts the following scenario:

Aircraft Deviating from Assigned Path: Show the aircraft positioned off its assigned path.

Nearby Aircraft on Expected Flight Paths: Display other aircraft on their designated paths nearby, indicating the spatial closeness to the deviating aircraft.

Other Aircraft Traffic: Show additional aircraft at different altitudes or distances, each on their own clear flight paths.

Background Elements: show different clouds."

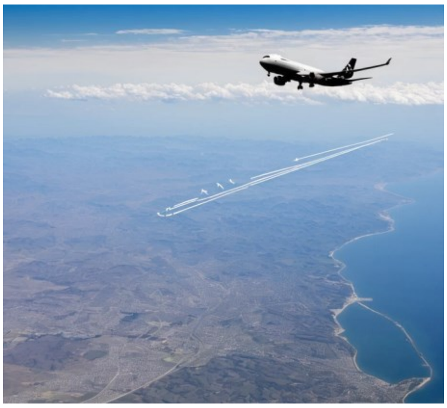
The prompt captures the apparent actual behaviour but does not precisely specify the interacting elements' purposes and how the behaviour is confusing. The quality of confusion is implicit within the explicit mention of the general rule used, and the architect considers it problematic. If re-used, the prompt may yield different results, but the concept remains conceptually isomorphic.

I.4.1.3 Step 1.3) Define the complex of the complexity field

In this step, we focused on defining the complex of the complexity field by analysing the visual scenario of the aircraft situation to identify interconnected elements contributing to the problem. This analysis aimed to enhance our understanding of how these components coexist and influence each other within the context of the situation. Guided by the architect's previous assertion about the visual representation, we applied general systems rules to recognise that any complexity comprises a collection of coexisting elements, whether they interact or not. We formulated a predictive question to determine the specific coexisting elements involved in the problematic situation as depicted in the model.

Following this, we created a guiding prompt to infer the complex of elements imagined to be part of the perceived scenario. To complete this step, we examined the visual representation alongside all previously discussed content to generate a comprehensive list of elements. The architect ultimately asserted that the following list accurately represents the complex of concerns: {ownship aircraft, by-passing aircraft, other aircraft, clouds, terrain (land), sea}. This structured list encapsulates the critical components contributing to the observed complexity, ensuring that all relevant systems are acknowledged while avoiding any implausible or omitted elements in the context of the scenario.

Table I.3 Defining the complex of complexes in problematic train track zone scenarios

SECoT Title	The complex of complexes definition
Input	**Architect assertion 1.2.4:** The architect asserts that the following depiction model captures the confusing behaviour faithfully: 
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 Observer). It involves an operational environment of coexisting complexes and the complicated nature (or types) of their relationships and interactions (epistemic uncertainty) for a Predictive Observer to predict their past, present, and future situations (managing aleatoric uncertainty or randomness).

Predictive Thinking Method	**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: The following list represents the complex of concerns: {ownship aircraft, by-passing aircraft, other aircraft, clouds, terrain (land), sea} **ownship aircraft:** means the observing aircraft which deploys the system of interest. We reused the term from the original paper of this case study.

The output is considered acceptable when the output list contains all systems. It is unacceptable if the list includes elements not seen in the depiction or implausible in the specified situation's complexity. Also, there are missing elements that should be mentioned in the list.

Implementation: Examine the depiction and all content mentioned in the analysis so far and generate a list of elements.

I.4.1.4 Step 1.4) Define the supra-complexes and their PrimePs

In this step, we focused on defining the supra-complexes that encompass the previously identified concerns. This involved analysing the problematic components to determine the overarching systems they belong to. We began by clarifying the primary purpose of each component, which is essential for understanding the goals each element needs to achieve in alignment with its Prime Purpose (PrimeP). Utilising general systems rules, we posed a predictive question to identify the interacting supra-complexes relevant to the coexisting elements. A guiding prompt encouraged us to classify the identified systems as parts of larger supra-complexes with shared purposes.

Through this analysis, we established three key supra-complexes: "Ownship Aircraft," which includes the {ownship aircraft}; "Other Aircraft," encompassing {by-passing aircraft, other aircraft}; and "Environment," which incorporates {clouds, land terrain features, sea features}. The architect confirmed that this classification accurately represents the relationships among the various components within the problem domain, clarifying how these elements coexist and interact within their respective contexts.

Table I.4 Definition of supra complexes in train network and adversarial scheme context

<table>
<tr><td>SECoT Title</td><td>Supra-complexes and their PrimePs definition</td></tr>
<tr><td>Input</td><td>Architect assertion 1.3.4: The architect asserts that:

The following list represents the complex of concerns:
{ownship aircraft, by-passing aircraft, other aircraft, clouds, terrain (land), sea}</td></tr>
<tr><td>General Systems Rules</td><td>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</td></tr>
<tr><td>Predictive Thinking Method</td><td>Predictive question 1.4.1a: What are the observed interacting Complexes, the possible holistic PrimeP they serve, and their Primary capability (or function) 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.</td></tr>
</table>

	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 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.
Output Prediction	**Architect assertion 1.4.6:** Ownship aircraft, which includes {Computerised perception-based mid-air collision avoidance system (AVP), Ownship aircraft subsystems} Other aircrafts, which include {by-passing aircraft, other aircraft} Environment, which includes {clouds, land terrain features, sea features} The architect asserts the PrimeP definitions for each Supra Complex as follows: 1. **Ownship Aircraft PrimeP:** Maintain safe operational flight and avoid mid-air collisions. 2. **Other Aircraft PrimeP:** Operate within shared airspace, adhering to established flight paths and protocols for collision avoidance. 3. **Environment:** neutral purpose that impacts the shared airspace between Ownship Aircraft and Other Aircrafts.

The output is accepted when a list of Supra Complexes, and their parts is defined. The supra complexes' naming convention should be capitalised.

Implementation: We consider three supra complexes:

Ownship aircraft, which includes {ownship aircraft}

Other aircrafts, which include {by-passing aircraft, other aircraft}

Environment, which includes {clouds, land terrain features, sea features}

We evaluated each element within its complex to determine the PrimeP (primary purpose) for each Supra Complex. We defined the overarching purpose that best captures its essential function and intent within the operational context. We focused on ensuring that each PrimeP provides clear, consistent goals across all scenarios, aligning with General Rule D's requirement for a stable and comprehensible purpose across observers and contexts. This alignment helps ensure that the priorities of the system components remain consistent in guiding design and operational decisions.

1. **Ownship Aircraft:** The purpose of this complex is primarily guided by the function of the AVP (computerised perception-based mid-air collision avoidance system) and Ownship aircraft subsystems, both of which are dedicated to maintaining safe flight operations and preventing mid-air collisions. Given these elements, we define the PrimeP of the Ownship Aircraft complex as: **"Maintain safe operational flight and avoid mid-air collisions."** This purpose reflects the primary safety and operational concerns of the Ownship's systems and aligns with the fundamental goal of safely navigating airspace.
2. **Other Aircraft:** The Other Aircraft complex includes potential by-passing aircraft and other aircraft in the vicinity. This Supra Complex's function is implicit in the role of aircraft within shared airspace and is defined relative to Ownship. The PrimeP for Other Aircraft is thus to **"Operate within shared airspace, adhering to established flight paths and protocols for collision avoidance."** This purpose supports system cohesion by setting clear expectations for cooperative, non-collision behaviour in shared airspace.
3. **Environment:** The Environment Supra Complex consists of non-aircraft elements, such as clouds, terrain, and sea features, that indirectly impact navigation and collision avoidance by shaping situational awareness and visibility. However we wont consider a PrimeP for it.

I.4.1.5 Architect Prediction 1.5

Given the above list of thinking steps, architect prediction entails the following definition of problem domain complexity:

Table I.5 Architect output prediction

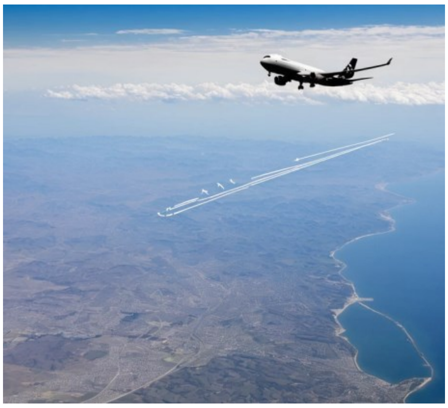
Architect Prediction
Architect assertion 1.1.4: **Confusing and Unsafe Behaviours Identified:** 1. **Aircraft Deviating Unexpectedly from Assigned Flight Path:**

- **Description:** An aircraft deviates from its assigned path, intersecting with a nearby flight path in the process.
- **Unsafe Aspect:** This behaviour contradicts expected adherence to assigned routes, introducing ambiguity about the aircraft's intentions and position for other pilots and ATC.
- **Undesirable Outcome:** This unplanned deviation could increase collision risk as nearby aircraft or ATC might not anticipate this course change, leading to unsafe proximity or even a direct collision.

2. **By-passing Aircraft Moving Erratically at a High Altitude near Crowded Airspace:**
    - **Description:** An unauthorised or unplanned aircraft enters high-altitude airspace without coordinated communication, moving in a way that suggests erratic or unpredictable flight patterns.
    - **Unsafe Aspect:** Erratic movement within controlled airspace is unexpected and creates difficulty for ATC in predicting the bypassing's path, especially if the aircraft does not respond to communication.
    - **Undesirable Outcome:** This unpredictability can obstruct ATC's ability to route nearby aircraft safely and may cause surrounding aircraft to take evasive actions, increasing the likelihood of incidental close calls or collision.

Architect assertion 1.2.4:

The architect asserts that the following depiction model captures the confusing behaviour faithfully:



Architect assertion 1.3.4: The architect asserts that:

The following list represents the complex of concerns:

{ownship aircraft, by-passing aircraft, other aircraft, clouds, terrain (land), sea}

Architect assertion 1.4.6:

Ownship aircraft, which includes {Computerised perception-based mid-air collision avoidance system (AVP), Ownship aircraft subsystems}

Other aircrafts, which include {by-passing aircraft, other aircraft}

Environment, which includes {clouds, land terrain features, sea features}

The architect asserts the PrimeP definitions for each Supra Complex as follows:

1. **Ownship Aircraft PrimeP:** Maintain safe operational flight and avoid mid-air collisions.
2. **Other Aircraft PrimeP:** Operate within shared airspace, adhering to established flight paths and protocols for collision avoidance.

I.4.2 Predictive Thinking Pipeline 2: Resolve the Complicatedness pattern of the observed complexity.

Predictive Thinking Pipeline 1 helped us clearly, objectively, and justifiably define the initial impression of the problem holistically. We exposed the overall structure of the problem domain complex. In this Predictive Thinking Pipeline, we will attempt to uncover the intricate interactions among the parts.

I.4.2.1 Step 2.1) Problem Interaction Analysis Using Actions Matrix

In this step, we systematically mapped out interactions between each identified system within the airspace conflict management environment by applying an Actions Matrix approach. This matrix captured relationships between the Ownship Aircraft, the AVP (Computerised Perception-Based Mid-Air Collision Avoidance System), By-passing Aircraft, Other Aircraft, and Environmental Factors (clouds, terrain, and sea).

Each system's influence over others was defined through binary interactions, detailing each action's role in guidance, avoidance, detection, sensing, or complication (physical and visual). By pairing each source complex with every other sink complex, we identified 24 key interactions, which were then articulated in a concise action format: [source system][action][sink system]. This provided a complete mapping of direct and indirect interactions supporting airspace navigation and collision avoidance, ensuring a comprehensive understanding of influences within the system and achieving clarity on potential complications in a structured, predictable manner.

Table I.6 Interaction Analysis of Problematic Coexisting Elements within the Train Network and Adversarial Scheme

SECoT Title	Problem Interaction Analysis
Input	Architect Prediction 1.5
General Systems Rules	**General rule F:** Complicatedness definition. **Complicatedness:** the predictability of a given observation by the predictive observer's approach to minimising their epistemic uncertainty. It is the Impact of complexes' behaviours (events in observation) on some predictive observers' confidence in making decisions to manage, use or interact with the observed complexity field. **General rule G:** Actions Matrix.
Predictive Thinking Method	**Predictive question 2.1.1:** What is the complete set of interactions among the defined complex within problem domain complexity? **Guiding prompt 2.1.2:** Apply the Actions Matrix method 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 Method 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]. **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: The following list defines the complicatedness of the problem domain complexity: **n1**: [AVP][guide][Ownship ac] **n2**: [By-passing ac][intersect][Ownship ac]

	n3: [Other ac][avoid][Ownship ac] **n4**: [Environment][Physically/visually complicate][Ownship ac] **n5**: [Ownship ac][sense][Environment] **n6**: [Ownship ac][govern][AVP] **n7**: [By-passing ac][Physically/visually complicate][AVP] **n8**: [Other ac][Physically/visually complicate][AVP] **n9**: [Environment][Physically/visually complicate][AVP] **n10**: [AVP][consider][Environment] **n11**: [Ownship ac][avoid][By-passing ac] **n12**: [AVP][detect][By-passing ac] **n13**: [Other ac][avoid][By-passing ac] **n14**: [Environment][Physically/visually complicate][By-passing ac] **n15**: [By-passing ac][sense][Environment] **n16**: [Ownship ac][avoid][Other ac] **n17**: [AVP][detect][Other ac] **n18**: [By-passing ac][avoid][Other ac] **n19**: [Environment][Physically/visually complicate][Other ac] **n20**: [Other ac][sense][Environment] **n21**: [Ownship ac][sense][AVP] **n22**: [AVP][detect][Environment] **n23**: [By-passing ac][sense][Ownship ac]

	n24: [Other ac][sense][By-passing ac]

The output is considered acceptable when;

1) All relationships are identified.
2) All relationships are briefly described in a single verbal phrase, no more than three words, and the nature of the problematic behaviours (actions). Only actions are defined.
3) Interactions are defined using the following format : [source system][action][sink system].

The output is considered unacceptable if:

1) There are missing interactions.
2) Interactions are defined using AIC factorisation method.
3) Interactions are not defined using the following format : [source system][action][sink system].

Implementation: Following are the steps of the Actions Matrix implementation:

1. **Identify Source and Sink Complexes**:
   - Source systems included Ownship Aircraft, AVP, By-passing Aircraft, Other Aircraft, and Environmental Factors.
   - Sink systems were similarly defined for all elements to capture the interaction and response relationships among all components.
2. **Define Action Types in Interaction Format**:
   - For each pair of interacting elements, a specific action was identified. Actions were briefly defined in terms of the specific interaction required, following the format [source system][action][sink system]. This captured functional relationships such as guidance, avoidance, detection, sense, and complication (physically or visually).
3. **Map All Binary Interactions**: By systematically pairing each source complex with each sink complex, a total of **24 binary relationships** were identified and defined in concise terms. This allowed for a holistic understanding of interactions, capturing both direct and indirect influences on flight safety.

The format of the Actions Matrix would include the following interactions:

Table I.7 Actions Matrix representing interactions among problematic coexisting elements in the complexity field

	Ownship ac	**AVP**	**By-passing ac**	**Other ac**	**Environment**
Ownship ac		n 6	n 11	n 16	n 21

AVP	n 1		n 12	n 17	n 22
By-passing ac	n 2	n 7		n 18	n 23
Other ac	n 3	n 8	n 13		n 24
Environment	n 4	n 9	n 14	n 19	
	Ownship ac	**AVP**	**By-passing ac**	**Other ac**	**Environment**
Ownship ac		govern	avoid	avoid	Sense and predict
AVP	guide		detect	Detect	Consider
By-passing ac	Intersect	Physically/visually complicate		avoid	Sense and predict
Other ac	avoid	Physically/visually complicate	avoid		Sense and predict
Environment	Physically/visually complicate	Physically/visually complicate	Physically/visually complicate	Physically/visually complicate	

I.4.2.2 Step 2.2) Predict the contributing factors (unsafe situations or opportunities)

In this step, we re-write the set of interactions we discovered that constitute complicatedness and categorised them into two contributing factors categories:

- Unsafe problematic activities
- Beneficial / Non-problematic activities.

We do so by examining every interaction and elaborating on its aspects and anticipated impact. Unsafe problematic activities are clear problems that, whatever design solution is chosen, will require further analysis.

Table I.8 Classification of unsafe and beneficial situations in train and adversarial drone interactions

Unsafe problematic activities	**Beneficial or non-problematic activities**
1. AVP: 1.2 n10: [AVP][consider][Environment] 1.3 n12: [AVP][detect][By-passing ac] 1.4 n17: [AVP][detect][Other ac] 1.5 n22: [AVP][detect][Environment] 1.6 n1: [AVP][guide][Ownship ac] **2. By-passing ac:**	**no beneficial situations identified**

2.2 n2: [By-passing ac][intersect][Ownship ac] 2.3 n7: [By-passing ac][Physically/visually complicate][AVP] 2.4 n15: [By-passing ac][sense][Environment] 2.5 n18: [By-passing ac][avoid][Other ac] 2.6 n23: [By-passing ac][sense][Ownship ac] **3. Other ac:** 3.2 n3: [Other ac][avoid][Ownship ac] 3.3 n8: [Other ac][Physically/visually complicate][AVP] 3.4 n13: [Other ac][avoid][By-passing ac] 3.5 n20: [Other ac][sense][Environment] 3.6 n24: [Other ac][sense][By-passing ac] **4. Ownship ac:** 4.2 n5: [Ownship ac][sense][Environment] 4.3 n6: [Ownship ac][govern][AVP] 4.4 n11: [Ownship ac][avoid][By-passing ac] 4.5 n16: [Ownship ac][avoid][Other ac] 4.6 n21: [Ownship ac][sense][AVP] **5. Environment:** 5.2 n4: [Environment][Physically/visually complicate][Ownship ac] 5.3 n9: [Environment][Physically/visually complicate][AVP] 5.4 n14: [Environment][Physically/visually complicate][By-passing ac] 5.5 n19: [Environment][Physically/visually complicate][Other ac]	

I.4.2.3 Step 2.3) Design problem selection

This step will review the original problem and re-evaluate it. By re-evaluation, we mean we want to be more definite in what set of sub-problems we want to further tackle and which problems we choose to leave. In this step, we will classify the output from step 2.2 into two categories and justify our choice:

- Problems within the design sphere of concern. We will use the following label "To be solved".
- Situations outside the design sphere of concern. We will use the following label "To be dropped".

The justification of the design problem sets out the initial high-level problems' context and motivation. The design team may contest some design problem choices, an important activity. Also, it opens up a door to regulators or design assurance teams to seek justifiable reasons why some identified issues are being dropped. At this stage, the design team can apply ethical considerations, which then go towards the trustworthiness case. The following is an example of defining problems:

Table I.9 Design Problem Selection

Interaction	Potential Concern	Decision	Elaboration or Justification
n1: [AVP][guide][Ownship ac]	AVP guiding the ownship may require precise alignment with real-time conditions and safety protocols.	**To be solved**	Guidance by AVP is essential to avoid collisions; ensuring accuracy and responsiveness is a key safety priority.
n2: [By-passing ac][intersect][Ownship ac]	Potential collision risk due to intersection.	**To be solved**	Intersecting ownship poses a direct risk; requires strategies for conflict detection and avoidance.
n3: [Other ac][avoid][directly risks]	Avoidance by other aircraft helps prevent mid-air collision.	**To be solved**	Coordinated avoidance measures contribute directly to collision prevention in shared airspace.
n4: [Environment][Physically/visually complicate][Ownship ac]	Environmental factors like weather or terrain may impair the ownship's visibility and operations.	**To be solved**	Environmental complications are a significant safety risk; addressing visibility and navigational challenges is critical.
n5: [Ownship ac][sense][Environment]	Ownship sensing environment helps to detect obstacles and avoid hazards.	**To be solved**	Essential for situational awareness and collision avoidance; directly

			related to safe navigation.
n6: [Ownship ac][govern][AVP]	Ownship's control over AVP might affect AVP’s collision avoidance capabilities.	**To be solved**	AVP’s operations must integrate with ownship controls to support safety and autonomy goals.
n7: [By-passing ac][Physically/visually complicate][AVP]	By-passing aircraft might obscure AVP's sensors, impacting detection and avoidance.	**To be solved**	Obscuration by other aircraft poses a direct risk to AVP’s detection capability; mitigation is required.
n8: [Other ac][Physically/visually complicate][AVP]	Other aircraft may obstruct AVP’s view, hindering detection abilities.	**To be solved**	Reducing visual interference by other aircraft is crucial to maintaining AVP detection effectiveness.
n9: [Environment][Physically/visually complicate][AVP]	Environmental elements (e.g., clouds, terrain) can obstruct AVP’s sensors.	**To be solved**	Environmental obstruction could lead to missed detections; mitigating this risk is critical for AVP reliability.
n10: [AVP][consider][Environment]	AVP must evaluate environmental conditions for safety.	**To be solved**	Essential for real-time responsiveness, environment-aware AVP supports safe manoeuvring.
n11: [Ownship ac][avoid][By-passing ac]	Necessary for ownship to avoid by-passing aircraft to prevent collisions.	**To be solved**	Direct collision avoidance by ownship is a safety-critical function, necessitating further design focus.
n12: [AVP][detect][By-passing ac]	AVP’s detection of by-passing aircraft helps manage proximity and prevent accidents.	**To be solved**	The detection of by-passing aircraft by AVP is foundational for collision avoidance, and it must be optimised.
n13: [Other ac][avoid][By-passing ac]	Other aircraft avoiding by-passings reduces potential collisions.	**To be solved**	Coordination in avoidance of manoeuvres is key to safe airspace management and needs reinforcement.
n14: [Environment][Physically/visually complicate][By-passing ac]	Environmental factors may hinder by-passings’ awareness, potentially impacting safe distance maintenance.	**To be solved**	Environmental interference requires a solution to ensure safe airspace awareness and collision avoidance.
n15: [By-passing ac][sense][Environment]	By-passing aircraft sensing the environment improves situational awareness.	**To be solved**	Environmental sensing by by-passing aircraft is essential for navigation and collision avoidance in crowded airspace.

n16: [Ownship ac][avoid][Other ac]	Critical for ownship to avoid other aircraft to ensure safe airspace sharing.	**To be solved**	Avoidance of other aircraft is fundamental to safe navigation; precise design is needed to prevent conflicts.
n17: [AVP][detect][Other ac]	Detection by AVP of other aircraft aids in avoiding collisions.	**To be solved**	AVP's detection supports safe coordination and is essential for ownship's situational awareness.
n18: [By-passing ac][avoid][Other ac]	By-passings avoiding other aircraft support airspace safety.	**To be solved**	Avoidance manoeuvres by by-passing aircraft are necessary to maintain safe airspace navigation and reduce risks.
n19: [Environment][Physically/visually complicate][Other ac]	Environmental conditions may obstruct other aircraft's sensors or visual capabilities.	**To be solved**	Environmental effects on other aircraft's navigation capabilities need addressing to support safe avoidance.
n20: [Other ac][sense][Environment]	Other aircraft sensing environmental conditions allows for informed avoidance and navigation.	**To be dropped**	Other aircraft's environmental sensing is beneficial but peripheral to ownship's AVP system design.
n21: [Ownship ac][sense][AVP]	Ownship sensing AVP helps ensure it receives AVP's guidance and avoids conflicts.	**To be solved**	It is essential to coordinate AVP and ownship actions, especially in dynamic conditions, to support safety.
n22: [AVP][detect][Environment]	AVP's detection of the environment supports responsive guidance for the ownship.	**To be solved**	Crucial for AVP to detect environmental factors for appropriate guidance adjustments.
n23: [By-passing ac][sense][Ownship ac]	By-passing's awareness of ownship aids in safe distance maintenance.	**To be solved**	Sensing by by-passing aircraft contributes to safe separation and prevents collision.
n24: [Other ac][sense][By-passing ac]	Sensing of by-passing by other aircraft supports mutual avoidance.	**To be dropped**	Beneficial but outside the design scope focused on ownship's AVP system; can be coordinated with external safety protocols.

The table above clearly categorises design focus areas based on interaction priority and safety relevance. By defining which interactions are "to be solved," we outline the design team's focus on critical safety and operational issues in the airspace, while interactions "to be dropped" are classified as lower-priority or peripheral to the AVP's direct operational requirements.

I.4.3 Predictive Thinking Pipeline 3: Predict the Emergence of AIC Complexity Field for Detailed Operational Scenario Articulation

Predictive Thinking Pipeline 2 helped us define the potential problems that we need to solve and the choices of problems to solve. In this pipeline, we will dive deeper into the AIC complexity of each interaction discovered and then identify the intricate relationships.

I.4.3.1 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} which we refer to as a situation. For example, if you define perception-based avoidance system (avp) as a factor, a situation needs to be mentioned with it, for instance, {active_avp} which means avp is actively influencing the ownship aircraft system. 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. See Figure I.1 for the abstract model of Forward-Feed Partial AIC-SECoT:

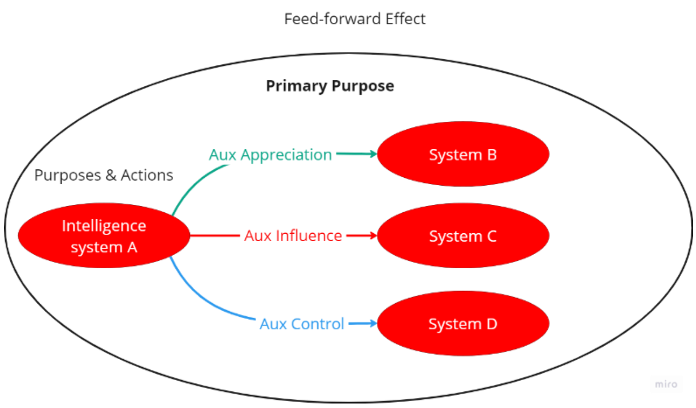


Figure I.1 Forward-Feed partial AIC-SECoT

In our case study, we will consider only the Forward-Feed partial AIC-SECoT. We will choose AVP guidance of ownship aircraft, which may lead to potential mid-air collision [derived from n1]. The key activity is we would do the AIC factorisation to each interaction in order to reveal the hidden interactions.

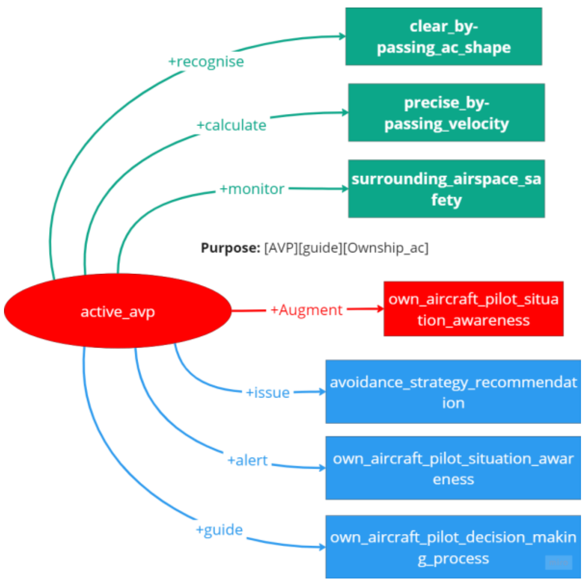


Figure I.2 Modelling AIC scenario from interaction n1

Figure I.2 illustrates the Active AVP Guidance System designed to enhance situational awareness and decision-making for the ownship aircraft pilot in scenarios that may lead to a potential mid-air collision. The model visually represents the interactions between the Active AVP system (red oval) and various situational parameters and decision-making processes through appreciation, influence, and control relationships.

Key Components and Interactions:

1. **Appreciation-based monitoring (Green Elements & Arrows)**
   - The **Active AVP** recognises, calculates, and monitors crucial situational parameters that contribute to flight safety:
     - **Clear By-Passing Aircraft Shape:** The system recognises the spatial characteristics of nearby aircraft to ensure safe manoeuvring.
     - **Precise By-Passing Velocity:** The system calculates the optimal velocity required to avoid conflicting air traffic safely.
     - **Surrounding Airspace Safety:** The AVP monitors environmental conditions, ensuring situational awareness in the operational domain.
2. **Influence on Pilot Awareness (Red Element & Arrow)**
   - The **Active AVP augments the ownship aircraft pilot's situational awareness**, ensuring the pilot has an enhanced perception of the airspace environment, potential hazards, and nearby aircraft.
3. **Control-based decision Support (Blue Elements & Arrows)**
   - The AVP actively engages with the **pilot's decision-making processes** by issuing guidance, alerts, and recommendations:

- **Avoidance Strategy Recommendation:** The AVP issues strategic recommendations to optimise collision avoidance.
- **Own Aircraft Pilot Situation Awareness:** The AVP alerts the pilot to changes in the operational environment.
- **Own Aircraft Pilot Decision-Making Process:** The AVP provides guidance to assist the pilot in executing effective avoidance manoeuvres.

The primary purpose of this system is to facilitate AVP-guided navigation for the ownship aircraft to enhance safety and reduce mid-air collision risks. By integrating appreciation (green), influence (red), and control (blue) interactions, the system ensures a multi-layered approach to situational assessment and pilot support, ultimately contributing to safer and more efficient airspace operations.

I.4.3.2 Step 3.2) Predict the extended list of emergent AIC interactions scenarios

We then capture the interactions in the figure above using the following SECoT. Note that we use the interaction format of |{situation}_[action]_{situation}| to help us with automating the extraction of factors.

In Appendix K, Table K.1, we experimented with a slight variation on the input of the SECoT. Instead of taking one interaction at a time, we used all 24 interactions as one detailed description of Unsafe problematic activities. The reason is for efficiency. The architect can extend each interaction individually or use multiple interactions. Our experiment showed that doing one interaction at a time provides more systematic explainability of how each interaction was considered in the design. When we summed them up, the explainability may require more clarification of how each interaction contributed to the whole factorisation process. Nonetheless, the exercise also proved to be useful in discovering those long-tail Black Swan interactions. Below is an example of how we used a single interaction n1 to extend an AIC scenario:

Table I.10 AIC extended scenario for n1 Interaction SECoT definition

Step 1: Unsafe problematic activities	**n1**: [AVP][guide][Ownship ac]	
Step 2: Observed System (obs)	**Step 3: Observed Action**	**Step 4: supra source Primary Purpose**
active_avp, own_ac	Active_avp guides own_Input Behaviour	Maintain safe operational flight and avoid mid-air collisions
Step 5: Auxiliary Influence interaction	**Step 6: Auxiliary Control interaction**	**Step 7: Auxiliary Appreciation interaction**

|{active_avp}_[+augment]_{ own_aircraft_pilot_situation_awareness}|	|{active_avp}_[+issue]_{ avoidance_strategy_recommendation}| |{active_avp}_[+alert]_{ own_aircraft_pilot_situation_awareness}| |{active_avp}_[+guide]_{ own_aircraft_pilot_decision_making_process}|	|{active_avp}_[+monitor]_{ surrounding_airspace_safety}| |{active_avp}_[+calculate]_{ precise_by-passing_velocity}| |{active_avp}_[+recognise]_{ clear_by-passing_ac_shape}|
Step 8: Predicted Problem Domain Factors or Features (with repetition)		
Appreciation = ['active_avp', ' surrounding_airspace_safety', 'active_avp', ' precise_by-passing_velocity', 'active_avp', ' clear_by-passing_ac_shape'] **Influence =** ['active_avp', ' avoidance_strategy_recommendation', 'active_avp', ' own_aircraft_pilot_situation_awareness', 'active_avp', ' own_aircraft_pilot_decision_making_process'] **Control =** ['active_avp', 'avoidance_strategy_recommendation', 'active_avp', ' own_aircraft_pilot_situation_awareness','active_avp', ' own_aircraft_pilot_decision_making_process']		

Table I.10 presents a Systems Engineering Chain-of-Thought (SECoT) definition for interaction n1, focusing on how an Active AVP guides the Ownship aircraft (own_ac) within a complex operational environment. The table follows the SECoT structured interaction format (|{situation}_[action]_{situation}|) to facilitate automated extraction of factors, ensuring a systematic analysis of interactions within the complexity field.

Step 1: Defining Unsafe problematic activities

The table begins by identifying an **unsafe problematic situation** where the **Active AVP guides the Ownship aircraft (own_ac)**, leading to potential flight safety concerns. This scenario is designated as **n1: [AVP][guide][Ownship ac]**, emphasising the control exerted by the AVP on the aircraft's behaviour.

Steps 2-4: Observed System, Action, and Supra Source Purpose

- **Step 2 (Observed System):** The primary interacting components are the **Active AVP** and the **Ownship aircraft**.
- **Step 3 (Observed Action):** The **Active AVP directly guides the Ownship aircraft's behaviour**, shaping its response to flight dynamics.
- **Step 4 (Supra Source Primary Purpose):** The ultimate purpose of this interaction is to maintain safe operational flight and avoid mid-air collisions, ensuring controlled manoeuvring in dynamic airspace environments.

Steps 5-7: Auxiliary Influence, Control, and Appreciation Interactions

The table then categorises auxiliary interactions that describe how the AVP influences, controls, and appreciates elements within the operational complexity field:

- **Step 5 (Influence Interactions):**
    - The AVP augments the pilot's situational awareness to enhance decision-making.
    - It issues avoidance strategy recommendations, improving navigational responses.
    - It alerts the pilot to critical airspace safety conditions.
- **Step 6 (Control Interactions):**
    - The AVP monitors surrounding airspace safety, continuously evaluating risks.
    - It calculates precise by-passing velocity, optimizing aircraft manoeuvres.
    - It recognises clear by-passing aircraft shapes, ensuring accurate decision-making.
- **Step 7 (Appreciation Interactions):**
    - The AVP appreciates the role of surrounding airspace safety in determining flight adjustments.
    - It acknowledges the importance of precise manoeuvring calculations to ensure collision-free paths.
    - It factors in the shape and trajectory of bypassing aircraft when guiding ownship movements.

Step 8: Predicted Problem Domain Factors or Features

The final step identifies recurring problem domain factors across Appreciation, Influence, and Control categories, helping to classify key system dependencies:

- **Appreciation Factors:**
    - The AVP appreciates **surrounding airspace safety**, the necessity of **precise by-passing velocity**, and the recognition of **clear by-passing aircraft shapes**.
- **Influence Factors:**
    - The AVP influences **avoidance strategy recommendations**, **ownship pilot situational awareness**, and **ownship pilot decision-making processes**.
- **Control Factors:**
    - The AVP controls the **avoidance strategy recommendation process**, **ownship pilot situational awareness**, and **ownship pilot decision-making process** to ensure safe manoeuvring.

I.4.3.3 Step 3.3) Collate factors/situations (step 8 in the table)

Collate all factors in between the curly brackets {} and capture them in AIC lists. Include the following information types: Source, Sink, and Supra Systems. For example;

Appreciation = ['active_avp', ' surrounding_airspace_safety', 'active_avp', ' precise_by-passing_velocity', 'active_avp', ' clear_by-passing_ac_shape']

Influence = ['active_avp', ' avoidance_strategy_recommendation', 'active_avp', ' own_aircraft_pilot_situation_awareness', 'active_avp', ' own_aircraft_pilot_decision_making_process']

Control = ['active_avp', 'avoidance_strategy_recommendation', 'active_avp', ' own_aircraft_pilot_situation_awareness','active_avp',

' own_aircraft_pilot_decision_making_process']

I.4.4 Predictive Thinking Pipeline 4: Predict and Evaluate Problem Domain Factors and Assumptions.

I.4.4.1 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. active_avp [15]
2. stabilised_own_aircraft_dynamics [11]
3. own_aircraft_pilot_decision_making_process [10]

Those factors are the most influential factors on our perception of the problem. However, they are predictable factors; we want to discover those heavy-tail Black Swan factors.

Evaluation of factor number of attentive mentions

The fascinating finding in this analysis is that the number of attentive mentions distribution of the factors followed a power-law distribution, which is an expected trend for Black Swan events and heavy tail distributions. This trend line created an easily fitted curve, a power-law distribution curve with 88% fitness. This also indicates that the analysis produced a reasonable result, as the output predicted a sensible set of factors corresponding to an anticipated architect’s predictive attention distribution.

Table I.11 Sample of Predicted Factors Output

Factor label	Predicted Factor	Number of attentive mentions
1	active_avp	15

2	stabilised_own_aircraft_dynamics	11
3	own_aircraft_pilot_decision_making_process	10
4	by-passing_aircraft_position	10
5	own_aircraft_pilot_situation_awareness	7
6	own_aircraft_flight_path	7

The table shows the top 16 factors with the number of attentive mentions of being mentioned more than 3 times. Some of these factors are rather obvious for the architect. However, there are 88 factors that have been mentioned 3 times or less. All of which were the hidden Black Swan events of the problem domain (relative to our initial knowledge about the problem). See Figure I.3, which shows how the long tail of Black Swan events has exposed the problem domain.

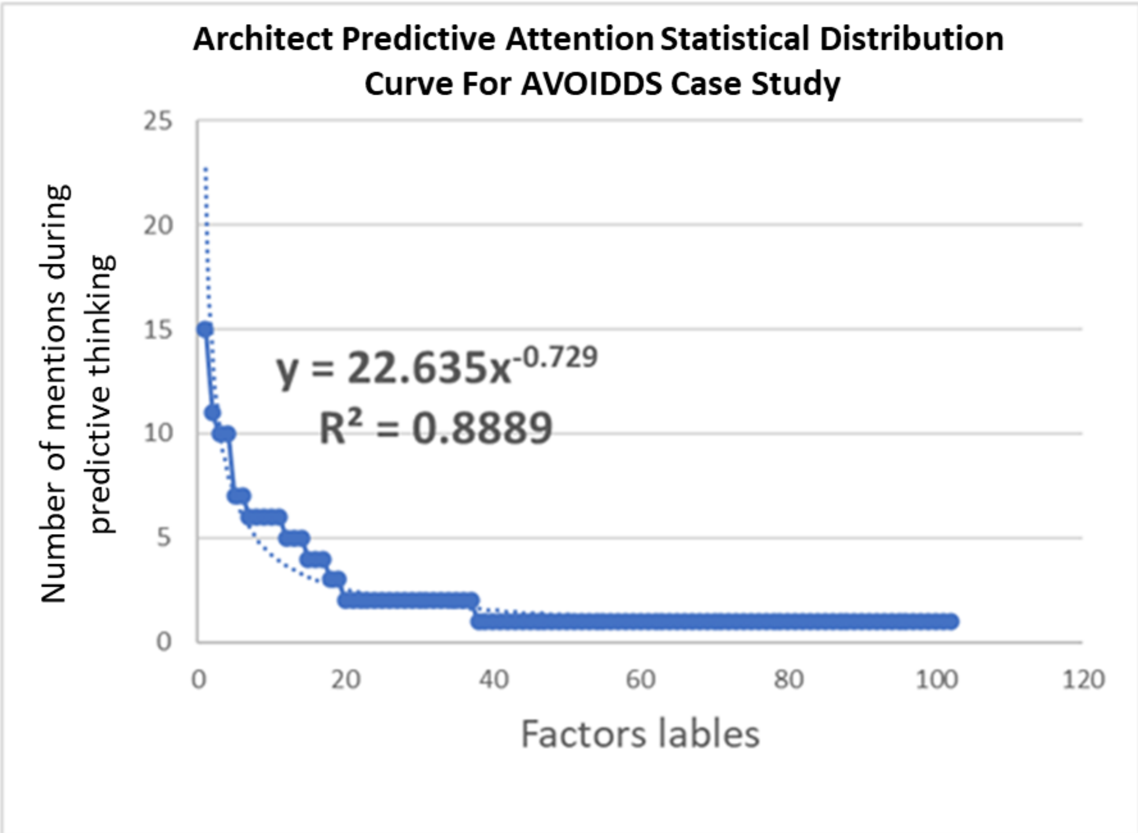


Figure I.3 Number of attentive mentions distribution curve of the derived factors

Table I.12 Full list of problematic factors to be considered for AVOID system problem domain

Factor label	Predicted Factor	Number of attentive mentions	Concern Level
1	active_avp	15	6.70%
2	stabilised_own_aircraft_dynamics	11	4.91%
3	own_aircraft_pilot_decision_making_process	10	4.46%
4	by-passing_aircraft_position	10	4.46%
5	own_aircraft_pilot_situation_awareness	7	3.13%
6	own_aircraft_flight_path	7	3.13%

7	surrounding_airspace_safety	6	2.68%
8	avoidance_strategy_recommendation	6	2.68%
9	visual_information	6	2.68%
10	by-passing_aircraft_speed	6	2.68%
11	collision_avoidance_system	6	2.68%
12	by-passing_aircraft_proximity	5	2.23%
13	conflicted_atc	5	2.23%
14	own_aircraft_flight_management_system	5	2.23%
15	collision_threat	4	1.79%
16	own_aircraft_camera	4	1.79%
17	other_aircrafts	4	1.79%
18	by-passing_aircraft_motion_pattern	3	1.34%
19	weather_data	3	1.34%
20	by-passing_aircraft_detection_tracking	2	0.89%
21	own_aircraft_roll_change	2	0.89%
22	own_aircraft_pitch_change	2	0.89%
23	own_aircraft_yaw_change	2	0.89%
24	own_aircraft_speed	2	0.89%
25	cloud_turbulence	2	0.89%
26	own_aircraft_flight_maneuvers	2	0.89%
27	rerouting_instructions	2	0.89%
28	pilot_stress	2	0.89%
29	by-passing_aircraft_visibility	2	0.89%
30	unpredictable_by-passing_aircraft_actions	2	0.89%
31	by-passing_aircraft_dynamics	2	0.89%
32	by-passing_aircraft_hack_avp_threshold	2	0.89%
33	pilot_alerting_and_warning_systems	2	0.89%
34	collision_threat_alert	2	0.89%
35	weather_conditions	2	0.89%
36	surrounding_airspace_obstructions	2	0.89%
37	own_aircraft_flight_path_change	2	0.89%
38	non_deterministic_intelligent_algorithms	1	0.45%
39	by-passing_aircraft_direction	1	0.45%
40	by-passing_aircraft_altitude	1	0.45%
41	threat_predictive_model	1	0.45%
42	risk_of_potential_collision	1	0.45%
43	cockpit_display_systems	1	0.45%
44	aircrafts_existing_warning_systems	1	0.45%
45	own_aircraft_altitude	1	0.45%
46	own_aircraft_radar	1	0.45%
47	own_aircraft_lidar	1	0.45%
48	environmental_obstructions	1	0.45%
49	fog	1	0.45%
50	raindrops	1	0.45%
51	cloud_cover	1	0.45%
52	cloud_type	1	0.45%
53	landscape_background	1	0.45%
54	daytime	1	0.45%
55	sunlight	1	0.45%

56	sun_position	1	0.45%
57	by-passing_aircraft_flight_path	1	0.45%
58	collision_time	1	0.45%
59	own_aircraft_response_time	1	0.45%
60	own_aircraft_pilot_feedback	1	0.45%
61	avp_real_time_training	1	0.45%
62	by-passing_aircraft	1	0.45%
63	emergency_directives	1	0.45%
64	unsafe_proximity	1	0.45%
65	airspace_conflicts	1	0.45%
66	radar_and_visual_range	1	0.45%
67	airspace_visibility	1	0.45%
68	wind	1	0.45%
69	avp_confidence_threshold	1	0.45%
70	own_aircraft_proximity	1	0.45%
71	autonomous_collision_avoidance_maneuvers	1	0.45%
72	potential_collision_threat_alert	1	0.45%
73	collision_avoidance	1	0.45%
74	automated_own_aircraft_flight_path_change	1	0.45%
75	own_aircraft_altitude_change	1	0.45%
76	disturbed_own_aircraft_dynamics	1	0.45%
77	own_aircraft_speed_change	1	0.45%
78	accelerated_own_aircraft_dynamics	1	0.45%
79	accelerated_own_aircraft_roll	1	0.45%
80	accelerated_own_aircraft_pitch	1	0.45%
81	accelerated_own_aircraft_yaw	1	0.45%
82	own_aircraft_flight_management_system_override	1	0.45%
83	manual_own_aircraft_flight_path_change	1	0.45%
84	weather_conditions_change	1	0.45%
85	weather_data_change	1	0.45%
86	own_aircraft_dynamics_change	1	0.45%
87	unpredictable_other_aircrafts_decision	1	0.45%
88	automated_collision_avoidance_maneuvers	1	0.45%
89	avp_detection_communication_delay_hack	1	0.45%
90	pilot_cognitive_limit	1	0.45%
91	surrounding_airspace_restrictions	1	0.45%
92	military_aircraft_intrusion	1	0.45%
93	other_aircrafts_presence	1	0.45%
94	bird	1	0.45%
95	drone_presence	1	0.45%
96	wind_shear_effect_on_camera	1	0.45%
97	air_density_change_due_to_high_speed_effect_on_camera	1	0.45%
98	aircraft_attitude_control	1	0.45%
99	mechanical_wear_and_tear	1	0.45%
100	autopilot_system_failure	1	0.45%
101	information_overload_for_pilot	1	0.45%
102	highly_stressfull_event	1	0.45%

I.4.4.2 Step 4.2) Define all identified problem domain factors.

Define each factor. We identify 102 types of factors involved within the complicatedness of the problem complexity. Below is the full definition for each factor (in no particular order);

1. **Own_aircraft_pilot_decision_making_process**: The process through which the pilot of the aircraft makes decisions based on various inputs, including alerts from the AVP system, situation awareness, and environmental conditions.
2. **By-passing_aircraft_position**: The location of an aircraft that is not part of the own aircraft's fleet, potentially posing a collision risk.
3. **Stabilised_own_aircraft_dynamics**: The situation of the own aircraft where its flight dynamics {e.g., speed, altitude, direction} are stable and under control.
4. **Avp (AVOID Dataset based Perception), active_avp**: A system designed to enhance the pilot's awareness and decision-making process by providing alerts and recommendations based on the surrounding environment and potential threats. active_avp means the avp system is actively influencing ownship aircraft.
5. **Own_aircraft_pilot_situation_awareness**: perception and comprehension of the current circumstances and surroundings of the aircraft by the pilot.
6. **Surrounding_airspace_safety**: the state of safety of the airspace surrounding the aircraft, taking into account environmental factors and the presence of other aircraft.
7. **Avoidance_strategy_recommendation**: The pilot receives recommendations from the AVP system on how to maneuver the aircraft to prevent possible collisions.
8. **Visual_information**: Information gathered from visual sensors, such as cameras, that provide pictures or videos of the area around the aircraft.
9. **Own_aircraft_flight_path**: The course or path that the aircraft is currently taking or intends to take.
10. **By-passing_aircraft_speed**: The speed of an aircraft passing by.
11. **Collision_avoidance_system**: An onboard system that detects and warns pilots of possible hazards in order to prevent collisions.
12. **Collision_threat**: An instance where there is a significant chance that the aircraft will collide with something, like another aircraft.
13. **By-passing_aircraft_proximity**: proximity or separation between the own aircraft and a by-passing aircraft.
14. **Conflicted_atc {Air Traffic Control}**: a service rendered by controllers stationed on the ground, who guide aircraft through controlled airspace. They are conflicted because, at any given moment, they are influenced by various priorities.

15. **Own_aircraft_flight_management_system**: An integrated computer system on board that controls important flight parameters.
16. **Other_aircrafts**: There are other aircraft nearby the one being used.
17. **By-passing_aircraft_detection_tracking**: The procedure for locating and tracking by-passing aircraft in real time.
18. **By-passing_aircraft_motion_pattern**: The movement pattern or behavior of a passing aircraft, including variations in direction and speed.
19. **Own_aircraft_camera**: Cameras installed on the aircraft that record visual data about the environment.
20. **Own_aircraft_roll_change**: Modifications to the aircraft's roll (or "lateral rotation") for maneuvering.
21. **Own_aircraft_pitch_change**: Modifications to the aircraft's pitch (or "vertical rotation") for maneuvering.
22. **Own_aircraft_speed**: The speed at which the aircraft is traveling.
23. **Own_aircraft_flight_maneuvers**: The aircraft's various manoeuvres to alter its speed, direction, or flight path.
24. **Rerouting_instructions**: Instructions from the flight management system or ATC to alter the current flight path for safety.
25. **Pilot_stress**: The pilot's degree of psychological strain or stress, especially during demanding or emergency situations.
26. **By-passing_aircraft_visibility**: The pilot's and the AVP system's capacity to visually identify a passing aircraft.
27. **Unpredictable_By-passing_aircraft_behaviour**: Unpredictable or irregular movements of a passing aircraft that make detection and avoidance tactics more difficult.
28. **By-passing_aircraft_dynamics**: The movement characteristics of the by-passing aircraft, such as its trajectory, speed, and altitude.
29. **Weather_data**: Weather-related data, including wind, visibility, and turbulence.
30. **Pilot_alerting_and_warning_systems**: Systems that notify the pilot of possible dangers or necessary actions.
31. **Collision_threat_alert**: An alert is sent out when a collision is imminent.
32. **Weather_conditions**: The weather conditions at the moment, including elements like wind, visibility, clouds, and turbulence.
33. **Surrounding_airspace_obstructions**: Environmental or physical barriers in the vicinity of the aircraft.
34. **Own_aircraft_flight_path_change**: Modifications to the aircraft's initial flight trajectory.

35. **Non_deterministic_intelligent_algorithms**: AVP system algorithms that produce results that are not solely based on their inputs are frequently employed in intricate decision-making procedures.
36. **By-passing_aircraft_direction**: The path or course that a passing aircraft is taking.
37. **By-passing_aircraft_altitude**: The altitude at which a passing aircraft is flying above the ground or sea level.
38. **Threat_predictive_model**: A model that uses recent data and trends to predict possible threats, like collision risks.
39. **Risk_of_potential_collision**: The possibility that the plane will crash into something else.
40. **Cockpit_display_systems**: Flight information, system statuses, and alerts are displayed on electronic display panels in the cockpit.
41. **Aircrafts_existing_warning_systems**: Existing aircraft systems that alert the pilot to potential dangers or necessary actions.
42. **Own_aircraft_yaw_change**: Modifications to the aircraft's yaw (horizontal rotation) for manoeuvring.
43. **Own_aircraft_altitude**: The altitude of the aircraft above sea level or the ground.
44. **Own_aircraft_radar**The aircraft's radar systems are used to identify nearby objects and other aircraft.
45. **Own_aircraft_lidar**: The aircraft's Lidar systems use laser light to illuminate the target in order to detect objects and measure distances.
46. **Environmental_obstructions**: Environmental or atmospheric factors that impair vision or sensor performance.
47. **Fog**: A weather pattern that makes it harder to see, which could affect sensor performance.
48. **Raindrops**: precipitation that may obstruct visibility and sensor accuracy.
49. **Cloud_cover**: The percentage of the sky that is cloudy, which can have an impact on sensor performance and visibility.
50. **Cloud_type**: Clouds are categorised according to their height and appearance, which affects visibility and flight conditions.
51. **Cloud_turbulance**: Flight stability is impacted by atmospheric turbulence linked to specific cloud types.
52. **Landscape_background**: The topography and background features that may have an impact on visibility and sensor readings.
53. **Daytime**: The time of day, which influences lighting and, consequently, sensor performance.
54. **Sunlight**: The sun's natural light affects visibility and how well visual sensors work.

55. **Sun_position**: The sun's position in the sky, which may affect pilot visibility and sensor performance.
56. **Own_camera**: The aircraft's installed cameras are used to record visual data.
57. **By-passing_aircraft_flight_path**: The path that a by-passing aircraft takes.
58. **Collision_time**: The approximate amount of time left before a possible collision.
59. **Own_aircraft_response_time**: The amount of time it takes for the aircraft to react to system commands or control inputs.
60. **Own_aircraft_pilot_feedback**: Pilot input or responses to alerts or system performance.
61. **Avp_real_time_training**: The ongoing process of updating and enhancing the AVP system in response to pilot feedback and real-time data.
62. **Emergency_directives**Critical events, like possible collisions, prompt the issuance of urgent instructions.
63. **Unsafe_proximity**: A distance that is deemed dangerously close between two aeroplanes.
64. **Airspace_conflicts**: circumstances where several aircraft's paths cross, increasing the chance of a collision.
65. **Radar_and_visual_range**: The range of objects that can be identified visually or by radar.
66. **Airspace_visibility**: The airspace's clarity, which impacts the capacity to visually identify other aircraft.
67. **Wind**: Wind conditions in the atmosphere can have an impact on sensor performance and flight dynamics.
68. **Cloud_turbulence**Flight stability is impacted by atmospheric disturbances linked to specific cloud formations.
69. **Avp_confidence_threshold**: The degree of assurance with which the AVP system consistently functions.
70. **By-passing_aircraft_hack_avp_threashold**: An attempt to change or tamper with the parameters or thresholds of the AVP system.
71. **Own_aircraft_proximity**: The separation between the aircraft and other objects or aircraft.
72. **Autonomous_collision_avoidance_maneuvers**: actions taken by the aircraft's systems to prevent collisions without the pilot's input.
73. **Potential_collision_threat_alert**: When a collision is possible, a warning is given.
74. **Collision_avoidance**: Actions taken to avoid a collision.
75. **Automated_own_aircraft_flight_path_change**: Automatic modifications to the flight path made by the aircraft's systems.
76. **Own_aircraft_altitude_change**: Modifications to the aircraft's flying altitude.

77. **Disturbed_own_aircraft_dynamics**: Variations in the aircraft's flight characteristics brought on by outside influences or manoeuvres.
78. **Own_aircraft_speed_change**: Modifications to the aircraft's flying speed.
79. **Accelerated_own_aircraft_dynamics**: The aircraft's dynamics during acceleration and deceleration.
80. **Own_aircraft_yaw_change**: Modifications to the aircraft's yaw (horizontal axis rotation).
81. **Accelerated_own_aircraft_roll**: The aircraft's roll during acceleration.
82. **Accelerated_own_aircraft_pitch**: The aircraft's pitch when accelerating.
83. **Accelerated_own_aircraft_yaw**: The aircraft's yaw during acceleration.
84. **Own_aircraft_flight_management_system_override**: The pilot's ability to manually override the automated flight management system.
85. **Manual_own_aircraft_flight_path_change**: Pilot-made manual modifications to the flight path.
86. **Weather_conditions_change**: Weather variations that may have an impact on sensor performance and flight dynamics.
87. **Weather_data_change**: Changes to the weather data that is being processed and received.
88. **Own_aircraft_dynamics_change**: Variations in the own aircraft's flight characteristics brought on by a number of factors.
89. **Unpredictable_other_aircrafts_decision**: Unexpected or careless choices made by other nearby aircraft.
90. **Automated_collision_avoidance_maneuvers**: Automatic manoeuvres carried out by the aircraft's systems to prevent collisions.
91. **Avp_detection_communication_delay_hack**: Deliberate attempts to cause delays in the AVP system's detection and communication processes.
92. **Pilot_cognitive_limit**: The limitations of the pilot's ability to process information and make decisions under stress.
93. **Surrounding_airspace_restrictions**: Limitations or prohibitions in certain areas of the airspace, affecting flight paths.
94. **Military_aircraft_intrusion**: The presence of military aircraft in the airspace, which can affect civil aviation operations.
95. **Other_aircrafts_presence**: The presence of other aircraft in the vicinity of the own aircraft.
96. **Birds**: Birds activity that can pose a risk to aircraft, especially during cruise, takeoff and landing.
97. **Drone_presence**: The presence of unmanned aerial vehicles {drones} in the airspace, which can pose collision risks.

98. **Wind_shear_effect_on_camera**: The impact of sudden changes in wind speed or direction on the operation of onboard cameras.
99. **Air_density_change_due_to_high_speed_effect_on_camera**: The effect of changes in air density, especially at high speeds, on camera performance.
100. **Aircraft_attitude_control**: The system and process of managing the orientation of the aircraft in flight.
101. **Mechanical_wear_and_tear**: The degradation of aircraft components over time due to regular use.
102. **Autopilot_system_failure**: A malfunction or failure of the aircraft's autopilot system.
103. **Information_overload_for_pilot**: A situation where the pilot is presented with more information than can be effectively processed.
104. **Highly_stressfull_event**: An event or situation that places a high level of stress on the pilot, affecting decision-making and performance.

I.4.4.3 Step 4.3) Define the assumptions made about factors

To do so, list all predicted emergent interactions without repetition. Then, describe the associated assumed situations, specifying which aspects of the factors are being assumed and which other elements are not. Lateral Predictive Thinking Methods can be used to help with imagining those assumptions.

In the previous case study, we did not incorporate hazard analysis at this stage. However, at this point, we will introduce hazard analysis to consider safety and security from the outset. This hazard analysis will inform the stakeholders and the architect in the next stage, enabling them to make engineering judgments based on the identified hazards perspective.

For example,

Factor 1: Own_aircraft_pilot_decision_making_process

It is assumed that the pilot will never over-rely on the perception system's judgement, maintaining their normal vigilance during flight.

Factor 2: By-passing aircraft position

The AVP system is assumed always accurately to detect and track the Bypassing aircraft's position.

In this process, we also experimented with adding further steps at this stage and with what else we could use the derived factors for. For the full table, see Appendix K, Table K.2. This time we included further validation and hazards identification analysis. Whereby we examined whether each assumption is plausible, hazardous or not. The following are the steps that architects can use to perform an extended analysis at this stage:

1. **Identify Assumptions:**
   - Fill columns titled "Predicted Factor", "Definition".
   - Clearly define your assumptions related to each discovered factor.
     - **Example:** Own_aircraft_pilot_decision_making_process: The process through which the aircraft's pilot makes decisions based on various inputs, including alerts from the AVP system, situation awareness, and environmental conditions.
     - **Explicit Assumption:** The pilot will never over-rely on the perception system's judgment, which may reduce their normal vigilance during flight.
2. **Evaluate the Realism of Assumptions:**
   - Fill columns titled "So what? Assumption", "Plausibility", "Why is it plausible or not?".
   - Assess whether the assumptions are realistic based on practical and operational scenarios.
     - **Evaluation:** Is the latter a realistic assumption?
     - **Answer:** No.
     - **Reason:** When the system demonstrates reliable results, pilots will eventually trust it more and more, leading to over trust.
3. **Determine Relevance to the Problem:**
   - Fill columns titled "Concerning?", "Why should we be concerned or not?".
   - Decide if the identified assumption should be included in the problem analysis.
     - **Question:** Should it be part of the problem?
     - **Answer:** Yes.
     - **Reason:** Directly impacting flight safety.
4. **Identify Potential Hazards:**
   - Fill columns titled "Hazard?", "How and Why is it hazardous?".
   - Assess if the assumption could lead to potential hazards or safety concerns.
     - **Question:** Is there a potential hazard?
     - **Answer:** Yes.
     - **Reason:** Pilots may unconsciously over-rely on the system over time, reducing their own personal vigilance and thus impacting airmanship.

I.4.4.4 Step 4.4) Identify problematic Black Swan events

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.

I.5 Stage 2: Architect Intent and Autonomous Solution Needs Definition[2]

This process would be used during Business or Mission Analysis in common systems engineering processes. It also defines the Operational Concept (OpsCon) for autonomous systems design. Although we didn't include the assessment of alternative solutions in this section, the output solution characterisation can guide the performance criteria to evaluate alternative solutions. This process also includes: **Stakeholder needs concept** definition on the back of solving problematic situations.

One pillar of AIC Systems Theory is the requisite of setting a purpose to realise streamlining the process in any system. The design team and their experiences, represented by the term "architect", is a system which requires an ideal purpose to aim for. The prime purpose of our approach is engineering an Ideal Whole, where an ideal autonomous systems solution can be realised or facilitated. Setting such a purpose sets the architect engineering approach in the right path towards making more comprehensive and objective design decisions, as it unifies the design team's ontology and language under a servitude of common PrimeP. This alone can add weight towards a more effective Trustworthiness Case for

The architect's intent is also clearly defined at this stage. Architects define high-level objectives of what the autonomous systems should do and be validated to demonstrate. This is a collaborative stage between the architect and the wider stakeholders. A document is distributed among the stakeholders before a series of works HazTOPS dedicated to making decisions about the list of assumptions. Here is another design review gate where ethical considerations base decisions on what to do with the problem scope. In this PhD, ethical considerations processes have not been considered.

The method includes the following activities to be done for every interaction and assumption:

Input: the primary input knowledge for the information to be included in the document is:

- Interaction definition.
- AIC model schema.
- Definition of factors.
- Definition of assumptions.

Process: the following thinking activities are to be done:

---

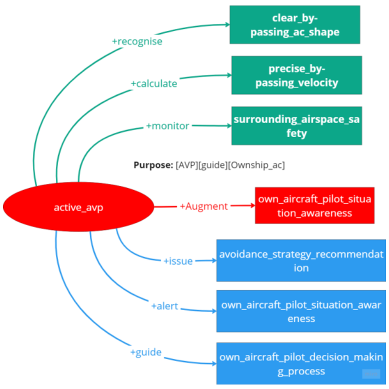
[2] For a summary see Section M.3

1. **Plausibility test:** is the assumption plausible or not?
2. **Why?:** define your justification on the reason behind the choice of plausibility value.
3. **Architect Intent (mission):** how does the architect intend to manage the situation? This may involve involving other stakeholders to manage this situation and not necessarily the system of interest
4. **Autonomous Solution needs:** What would be the objective of the autonomous systems in managing the situation?
5. **Other support systems need:** If the architect decides to involve other systems or stakeholders, what goals should they fulfil?
6. Define the Primary Purpose of the solution that the objectives of autonomous systems and other support systems are to fulfil. Define the overall purpose of the autonomous systems that fulfil part of PrimeP.

Output: The process's output is a definition of the overall solution strategy, including the Architect's Intent (mission), the autonomous systems and the support systems' needs and constraints. We call the template as "Architect High-Level Solution Prescription" or just Architect Prescription for short.

Autonomous systems goals represent the high-level validation objectives the system is expected to fulfil. Those objectives can guide the safety case construction regarding the highest-level goals and how the design and test achieved those goals, including supporting artefacts. Below is an application of the above step:

Table I.13 Architect High-Level Solution Prescription

<table>
<tr><td>Situation</td><td>By-passing_aircraft_dynamics: The characteristics of the By-passing aircraft's movement, including speed, altitude, and trajectory.
Assumption: It is assumed that the AVP system will not continuously and accurately track the dynamics of By-passing aircraft.

</td></tr>
<tr><td>Plausibility (Plausible/ Not plausible)</td><td>Plausible</td></tr>
</table>

Why?	Environmental conditions and sensor limitations can impact continuous and accurate tracking.
Architect Intent (mission)	The AVP training shall include variety of By-passing aircraft dynamics, at variety of speed.
Autonomous solution needs	The AVP shall continuously and accurately track the dynamics of By-passing aircraft in variety of environmental changes.
Autonomous solution constraints	Data Epistemic Coverage sufficiency.
Support Systems Needs	None. (In this case, we don't specify. However, there may be some relevant support systems that need to be considered.)
Support Systems Constraints	None. (In this case, we don't specify. However, some relevant support systems may need to be considered.)

I.6 Stage 3A: HazTOPS and Ordered AIC-driven Autonomous System Requirements Development[3]

I.6.1 Predictive Thinking Pipeline 1: Introducing Autonomous systems into Forward-Feed complexity

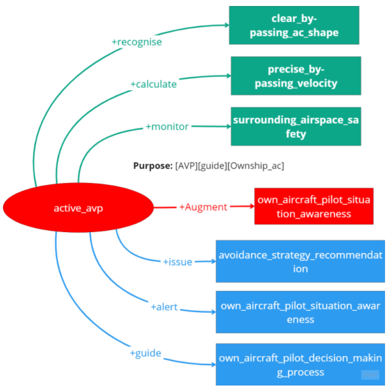
In this step, we derive the desired agent AIC interactions that fulfil the architect's intents and stakeholders' needs. The Predictive Thinking Pipeline strategy is defined using the following SECoT. We will use the following interaction example: Adversarial drone avoids vegetation around the fence,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

[3] For a summary see Section M.4

I.6.1.1 Step 1.1) Introduce the solution into the mix.

Model the AIC Schema and introduce the solution into the mix of the problem situation. In the train tracks problem, we used the interaction as an input: {roaming_adversarial_drone}_[-lead]_{train_derailment}, as well as the model. This time, for variety, we use the factors definition and the assumptions output, including the architect's intent and the autonomous system's purpose.

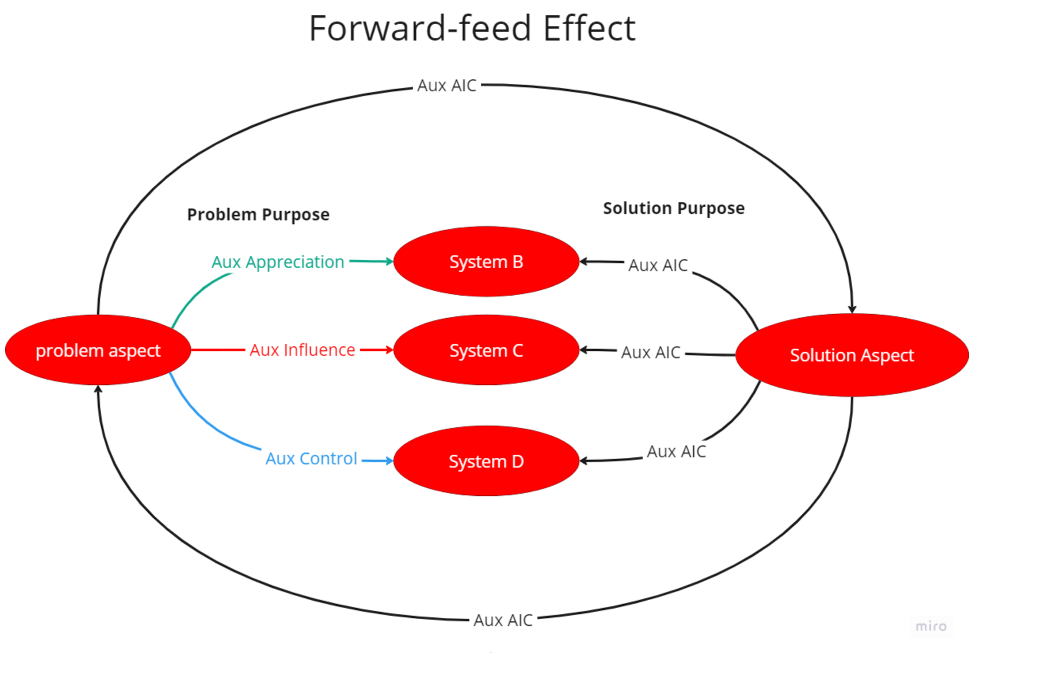
Table I.14 Implementing architect Intent and Forward-Feed AIC Interaction Framework for addressing AVP reliability for by-passing aircraft

<table>
<tr>
<td>Input</td>
<td>By-passing_aircraft_dynamics: The characteristics of the By-passing aircraft's movement, including speed, altitude, and trajectory.
Assumption: It is assumed that the AVP system will not continuously and accurately track the dynamics of By-passing aircraft.


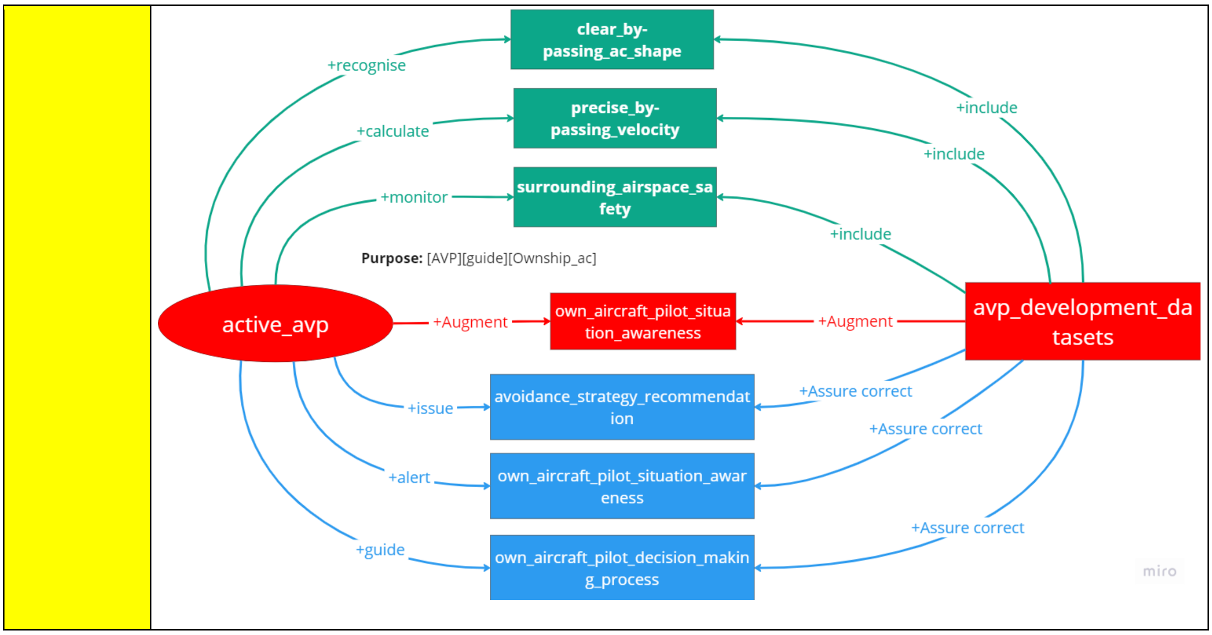
Architect intent:

The AVP training shall include a variety of By-passing aircraft dynamics at a variety of speeds.

Autonomous system purpose:
The AVP shall continuously and accurately track the dynamics of By-passing aircraft in various environmental changes.</td>
</tr>
<tr>
<td>General Systems Rules</td>
<td>General rule G: Emergence of AIC Complicated behaviour.
General rule H: Forward-Feed Effect AIC modelling schema.</td>
</tr>
</table>



Predictive Thinking Method	**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.
Output Prediction	**Architect assertion:** The architect asserts that:



The predicted solution mix includes the following interactions:

1. **Central AVP System Influence on Pilot Situation Awareness**

- **Influence Interaction (Red Arrows)**: The active AVP system (central red oval labelled "active_avp") influences the pilot's situational awareness. This is visualised by a red arrow labelled "+Augment," leading to the red rectangle labelled "own_aircraft_pilot_situation_awareness." This indicates that the AVP indirectly enhances or influences the pilot's understanding of the aircraft's situation and surroundings, thus improving situational awareness.

2. Control and Guidance of Pilot Support Functions

- **Control Interactions (Light Blue Arrows)**: The AVP exerts a control-like influence over three critical pilot support functions, represented by light blue rectangles:
  - **"avoidance_strategy_recommendation"**: The AVP system issues recommendations to the pilot, shown by a light blue arrow labelled "+issue."
  - **"own_aircraft_pilot_situation_awareness"**: In addition to augmenting the pilot's awareness indirectly, the AVP has a more direct, control-like effect here, alerting the pilot as necessary (light blue arrow labelled "+alert").
  - **"own_aircraft_pilot_decision_making_process"**: The AVP provides direct guidance to assist in decision-making, represented by a light blue arrow labelled "+guide."

These interactions suggest that the AVP actively governs or steers these aspects of pilot support, thus providing critical situational input and control over pilot responses to surrounding conditions.

3. Influence of AVP Development Datasets on Pilot Support Functions

- **Influence of Development Datasets (Light Blue Arrows)**: The AVP system relies on its development datasets (red rectangle labelled "avp_development_datasets") to ensure the accuracy and correctness of its pilot support functions. This relationship is depicted by light blue arrows labelled "+Assure correct", pointing from the development datasets to each of the three pilot support functions. This influence suggests that the datasets indirectly govern the precision and reliability of the AVP's recommendations, alerts, and guidance by shaping the AVP's capabilities during development.

4. Data-Driven Control Requirements for AVP System Operations

- **Control Requirements (Blue Arrows)**: The AVP system requires access to specific data considerations for its effective operation, depicted by blue arrows:
    - **"+recognise"**: The AVP must recognise "clear_by-passing_ac_shape" (blue rectangle), likely representing the shape or profile of nearby aircraft.
    - **"+calculate"**: The AVP calculates "precise_by-passing_velocity" (rectangle), reflecting the speed required for safe bypass manoeuvres.
    - **"+monitor"**: The AVP monitors "surrounding_airspace_safety" (blue rectangle), assessing environmental conditions and hazards.
    - These data-driven tasks are included in the AVP's development datasets (blue arrows labelled "+include"), indicating that these data considerations directly control and configure the AVP's functionality, shaping how it operates in real-world scenarios.

I.6.1.2 Step 1.2) Characterise the AIC interactions

Define the set of interactions between the sink and the source using AIC structured interaction format of: |{source situation}_[+,- or no sign, AIC-action]_{sink situation}|, Written in the following grammar: |{adjective+noun}_[verbal phrase]_{adjective+noun}| For example, |{flying_police_robot}_[learns humans' visual profiles]_{distressed_people}| Capture the output in the following table:

Table I.15 Mapping AIC interactions of AVP with ownship aircraft and the environment

Source: {active_avp}	
Output Behaviour	**Input Behaviour that impacts the emergence of Output Behaviour**
I1: |{ active_avp}_[+Augment]_ { own_aircraft_pilot_situation_awareness}|	**A1:** |{ active_avp}_[+monitor]_ {surrounding_airspace_safety}| **A2:** |{active_avp}_[+calculate]_ { precise_by-passing_velocity}| **A3:** |{ active_avp}_[+recognise]_ { clear_by-passing_ac_shape }|
	C1: |{ active_avp}_[+issue]_ {avoidance_strategy_recommendation}| **C2:** |{ active_avp}_[+alert]_ { own_aircraft_pilot_situation_awareness}| **C3:** |{ active_avp}_[+guide]_ { own_aircraft_pilot_decision_making_process}|
I2: |{ avp_development_datasets }_[+Augment]_{ own_aircraft_pilot_situation_awareness }|	**A4:** |{ avp_development_datasets}_[+include]_ {surrounding_airspace_safety}| **A5:** |{avp_development_datasets}_[+include]_ { precise_by-passing_velocity}| **A6:** |{ avp_development_datasets}_[+include]_ { clear_by-passing_ac_shape }|
	C4: |{ avp_development_datasets}_[+Assure correct]_ {avoidance_strategy_recommendation}| **C5:** |{ avp_development_datasets}_[+ Assure correct alert]_ { own_aircraft_pilot_situation_awareness}| **C6:** |{ avp_development_datasets}_[+ Assure correct]_ { own_aircraft_pilot_decision_making_process}|

Table I.15 presents a structured mapping of AIC interactions between the Active AVP and its surrounding operational environment. It focuses on its engagement with its ownship aircraft and the AVP development datasets. This table formalises the critical functional dependencies that shape AVP behaviour, ensuring enhanced situational awareness, decision-making, and strategic response formulation in dynamic airspace environments.

The first interaction category (I1) examines how the Active AVP directly influences the Ownship Pilot's situational awareness by augmenting it through key appreciation and control mechanisms. The appreciation behaviours (A1-A3) include monitoring the surrounding airspace safety, calculating precise by-passing velocity, and recognising clear by-passing aircraft shapes.

The AVP is guaranteed to have a thorough understanding of the current flight conditions thanks to these appreciation interactions. The control behaviours (C1-C3) focus on issuing avoidance strategy recommendations, alerting the pilot to potential risks, and guiding the pilot's decision-making process, ensuring that the AVP perceives and actively intervenes in flight safety operations.

The second interaction category (I2) emphasises how structured data-driven enhancements from AVP development datasets can enhance the ownship pilot's situational awareness. The appreciation interactions (A4-A6) emphasise the inclusion of critical environmental parameters such as surrounding airspace safety, precise velocity calculations, and clear aircraft shape recognition in training datasets. This ensures that the ML components within the AVP are trained on diverse and comprehensive operational scenarios. The control interactions (C4-C6) reinforce the necessity of assuring the correctness of avoidance strategy recommendations, alert protocols, and pilot decision-making processes, ensuring that the AVP's actions remain aligned with validated safety objectives.

I.6.2 Predictive Thinking Pipeline 2: Designing the affecting Backward-Feed complexity field

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

I.6.2.1 Step 2.1) Visualise the operational design domain environment

Visualise the operational design domain environment with the Complicated Behaviour being part of it. To start modelling the operational environment, we need a real-world picture of the theatre of operations to help with specifying the operational environment complexity field. Figure I.7 captures a typical scenario faced by the AVP system (mid-air collision avoidance perception system).

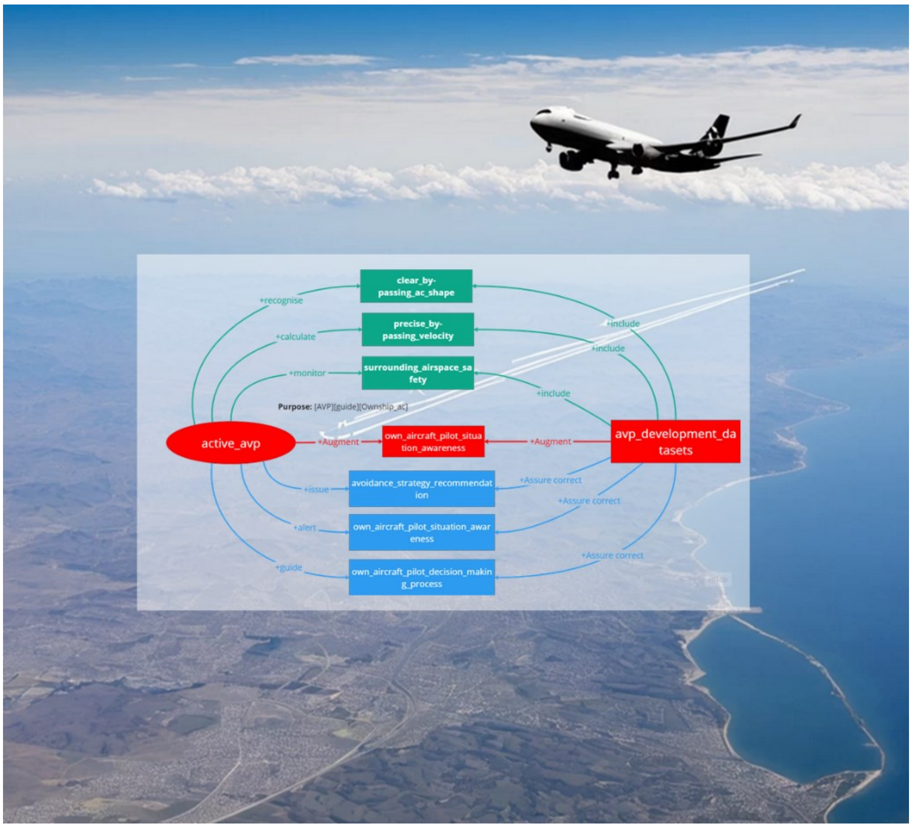


Figure I.4 A view synthesized using a message to image transformer of a typical scenario might be faced by the AVP.

The model captures the interconnectedness between these immediate and remote factors to produce a comprehensive dataset that supports a resilient and adaptable AVP system. The kinds of images and scenarios included in the training dataset are directly influenced by immediate factors like the weather, the surrounding landscape, and the kinds of objects in the area. In the meantime, distant variables such as long-term climate change indirectly impact the training data's relevance over time by progressively changing the patterns and number of attentive mentions of these immediate conditions.

The goal of this model is to capture the intricate relationships between variables that influence and are influenced by the training datasets utilised to create an AVP (Aircraft Vision-Based Avoidance Detection) system. Building a solid dataset will help the AVP system recognise and steer clear of a variety of objects and obstacles while in flight, ensuring dependable and secure operation in a range of dynamic circumstances. These factors are divided into two primary groups by the model:

1. **Remote Affecting & Affected Problem Complex**
   - **Long-Term Climate Change Shift**: the term refers to gradual but noteworthy modifications in climatic patterns over time, which could change the kinds,

number of attentive mentions, and intensity of environmental conditions that the AVP system must manage. By adding new scenarios that might not be adequately represented in current datasets, these changes have an effect on the training data's long-term relevance and coverage. For instance, the AVP system needs to be ready to manage increasingly frequent but historically uncommon conditions as climate change changes bird migration patterns or increases the number of attentive mentions of extreme weather events like storms and heavy fog. Thus, long-term climate change indirectly affects the focus and makeup of training data by posing new environmental problems that have an effect on flight safety.

2. **Immediate Affecting & Affected Problem Complex:**

This group defines the variety and range of scenarios that the model is exposed to and includes elements that have a more immediate and direct effect on the AVP training datasets. Making sure the AVP system is capable of recognising and avoiding a variety of objects, weather patterns, and other pertinent environmental factors is the aim.

- **Weather Conditions Variety Definition**: The performance of the AVP system and visibility can be greatly impacted by weather conditions like wind, rain, fog, and snow. To assist the model in learning to recognise objects under various visibility and sensor performance conditions, training datasets must encompass a broad range of weather scenarios.
- **Landscape Types Variety Definition**: The visual context in which objects appear varies depending on the type of landscape, including desert, coastal, urban, and mountainous areas. Diverse landscape backdrops should be included in the training data to improve situational adaptability because the AVP system should be able to function well in a variety of terrains.
- **Geographical Region-Specific Natural Phenomena**: It is crucial to incorporate regional natural phenomena like sandstorms in deserts, blizzards in polar regions, or clouds of volcanic ash close to active volcanoes. These particular phenomena have the potential to impact avoidance behaviours and present special visibility challenges.
- **Birds**: Birds are a frequent and unpredictable source of collisions. For the AVP training datasets to enhance the system's detection and avoidance of avian obstacles, bird density, species diversity, and typical flight paths must be taken into consideration, as bird presence and behaviour vary by region and season.
- **Illumination Variety Definition**: Different lighting conditions, including dawn, dusk, night, and bright midday sun, can affect how objects appear to vision

sensors. Training data must represent various illumination conditions to help the AVP system recognise objects under varying lighting conditions and avoid false positives or missed detections due to shadowing, glare, or low light.

- **Aircraft Types Variety Definition**: The presence of different types of aircraft, such as helicopters, small private planes, commercial jets, and drones, is crucial for collision avoidance. Each type has distinct shapes, sizes, and movement patterns that the AVP system should learn to identify, classify, and respond to accurately.
- **Other Flying Objects**: Besides aircraft, other flying objects such as balloons, recreational drones, or even kites can enter airspace, especially in urban or event-filled areas. The AVP training datasets should include diverse flying objects to handle these less predictable encounters.
- **Other Aircrafts Jet Stream**: Jet streams created by other aircraft can cause turbulence that affects AVP reliability. Recognising and anticipating jet streams indirectly impacts the AVP system by informing the pilot or autonomous system of potential turbulence regions. Training data should thus include scenarios involving other aircraft's proximity and behaviour.
- **Sea Vessel Types Variety Definition**: For low-flying aircraft or drones, sea vessels such as cargo ships, yachts, and fishing boats may occasionally enter the AVP system’s visual range, especially in coastal areas or during over-water flights. Identifying and avoiding these vessels is essential for safe, low-altitude operations.

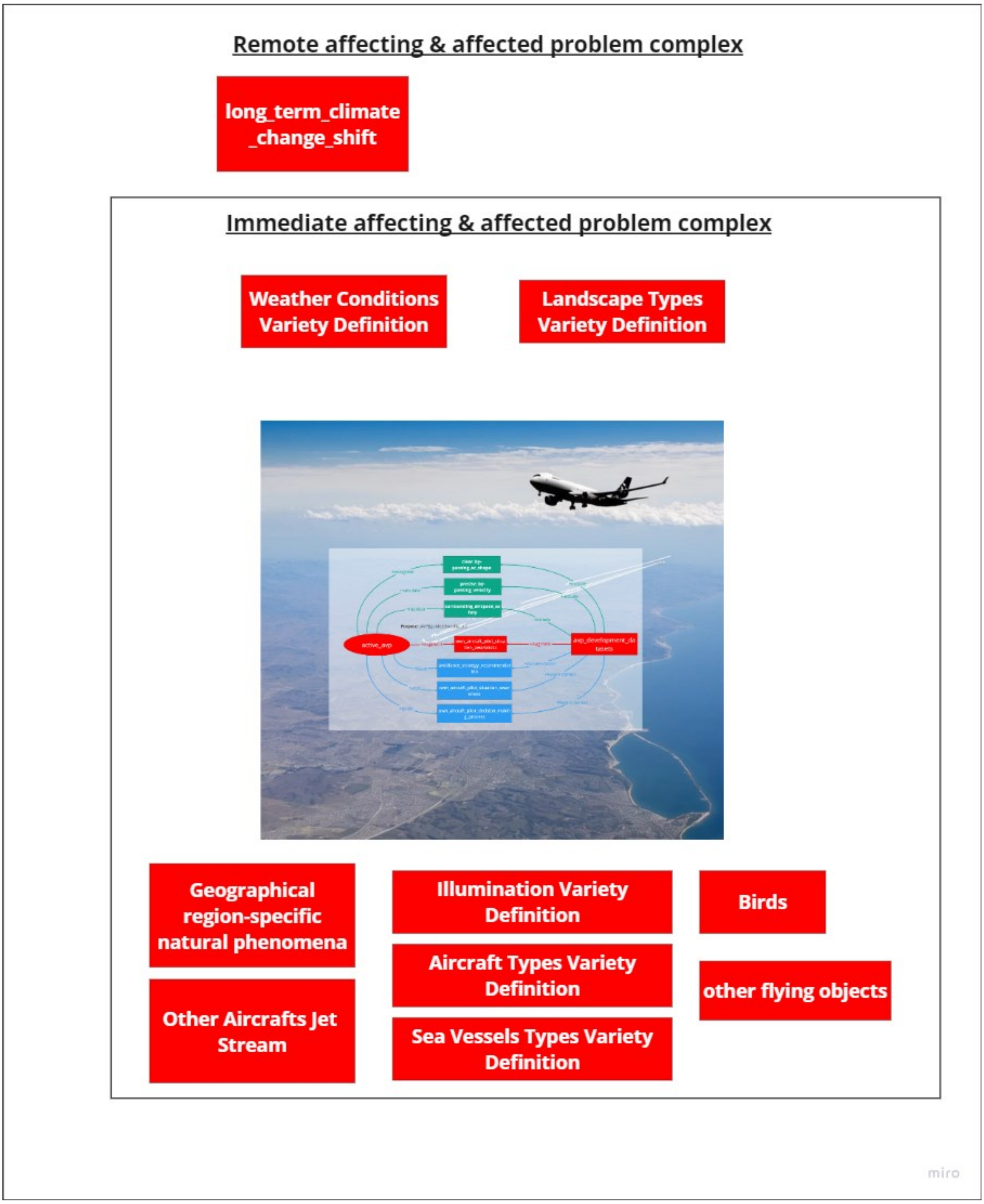


Figure I.5 Perception-based mid-air collision avoidance system operational environment

I.6.2.2 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 need to abstract the Forward-Feed model for ease of modelling. We reduce the model into a controlled language of interaction:

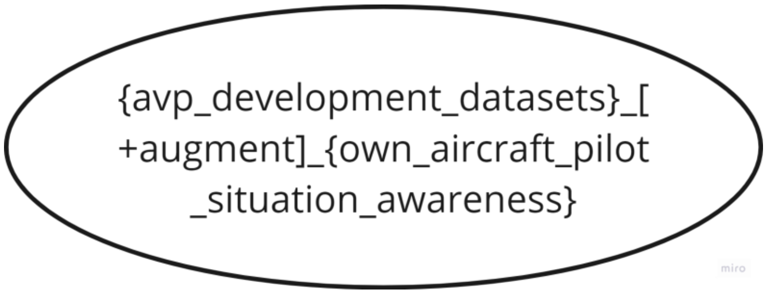


Figure I.6 I2 interaction

Then, we model the Backward-Feed AIC interactions by asking the following questions:

- What complex could apreciate the interaction? **We answer:**
  - The informed air traffic controller appreciates the AVP development dataset.
- What complex could influence the interaction? **We answer:**
  - Other flying objects could influence the AVP dataset development.
- What complex could control the interaction? **We answer:**
  - Weather conditions definitely control the AVP development dataset.
  - Architect a predictive mental model that can also control the quality of the AVP dataset coverage.

We will use icons to represent the complexes visually. Imagination is key enabler for predictive thinking. I.7 models captures the unknown unknowns at this stage which are also part of the possible long tail Black Swan Scenarios.

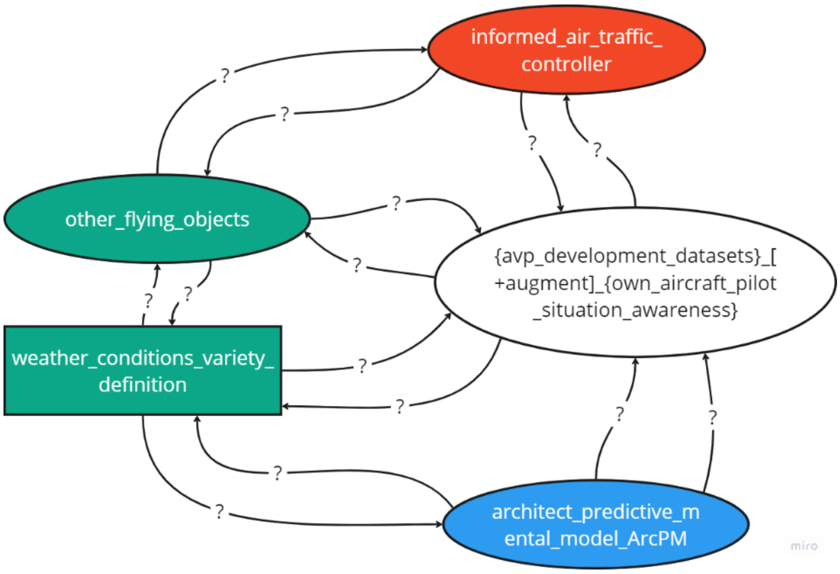


Figure I.7 Backward-feed model for I2 unknown unknown interactions. Black Swan scenarios.

I.6.2.3 Step 2.3) Comprehensively appreciate the complicatedness of the problem domain

To comprehensively define the complicatedness of the above complexity field abstraction, you may use the Actions Matrix method to define all the actions and their effect among the complexes. Note some actions do not make sense. Some of its actions are unsigned. The complexity involves the following complex of complexes:

Table I.16 Icon definition

Situation	Icon
other_flying_objects	
weather_conditions_variety_definition	
accurate_architect_predictive_mental_model_ArcPM	
informed_air_traffic_controller	
avp_development_datasets	

With the above table in mind, we can anticipate the following potential unknown interactions below. There should be 20 possible unknown interactions, for example (we will only list 10 of them):

1. |{ avp_development_datasets}_[?]_{ architect_predictive_mental_model_ArcPM}|
2. |{ avp_development_datasets}_[?]_{ other_flying_objects}|
3. |{ avp_development_datasets}_[?]_{ weather_conditions_variety_definition}|
4. |{avp_development_datasets}_[?]_{reliable_correct_target_object_recognition_confidence_scores}|
5. |{ avp_development_datasets}_[?]_{informed_air_traffic_controller}|
6. |{avp_development_datasets}_[?]_{visual_traffic_density_and_patterns}|
7. |{avp_development_datasets}_[?]_{critical_data_accuracy}|
8. |{ other_flying_objects}_[?]_{ avp_development_datasets }|
9. |{other_flying_objects}_[?]_{weather_conditions}|
10. |{other_flying_objects}_[?]_{other_flying_objects _visual_complexity_parameters}|

In order to ensure coverage, we will need to perform Action Matrix method. Table I.17 , below, presents an AIC (Appreciation, Influence, Control) coloured action matrix, mapping the interactions of the AVP dataset with key situational elements from Table I.16 (Icon Definition). The matrix highlights how the AVP dataset captures, predicts, informs, or complicates interactions across various domains, including other flying objects, weather conditions, air traffic controllers, predictive mental models, and AVP development datasets. And the following complicatedness (we introduced AIC colour scheme to help with constructing the AIC complexity field model):. Green represents “Appreciation”, Red represents “Influence” and Blue represents “Control”. In order for us to have a complete understanding we can also use the AIC modelling schema.

Table I.17 AIC coloured action matrix to model all possible interactions of the AVP dataset influence over

					
		+capture	+capture	+assure	+assure
	-visually complicate		visually complicate	visually complicate	-operationally complicate
	-visually complicate	operationally complicate		visually complicate	- operationally complicate
	+predict	+predict	predict		+assure
	+inform	+monitor	monitor	+inform	

I.6.2.4 Step 2.4) Complete the extended view of the AIC mental model schema

The Actions Matrix above defines how the complexity of the scenario is resolved. Now, we will transfer all the knowledge to the final output of the step, which is a comprehensive AIC mental model of the problem. Note how we started and how we ended. Every step in this process carefully explains how we produced the scenario. Note that the interactions between the Forward-Feed bubble and the Backward-Feed operational environment concern the source. We are currently interested in the AVP development datasets in the design. Later, if the design team wishes to evaluate the complexity from the sink perspective, the complexity will change. The approach is flexible and allows for investigating and modelling the whole from any node, thus creating a variety of complexities. Figure I.8 captures all the resolved interactions in I.17,

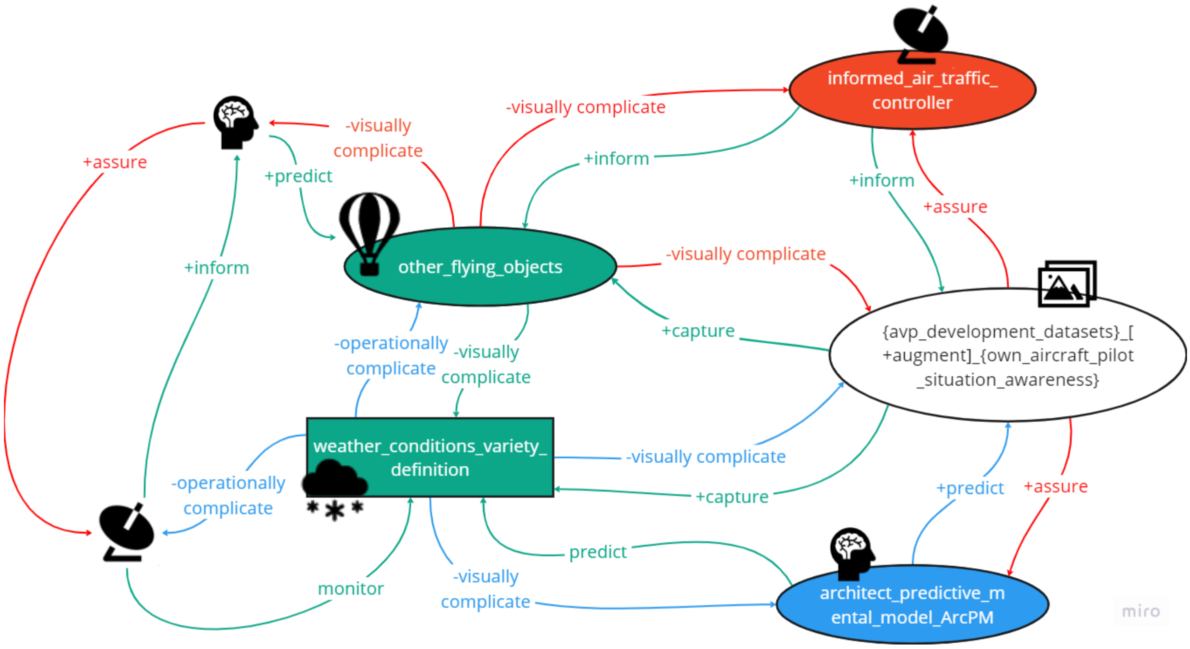


Figure I.8 Resolved complicatedness of AVP dataset complexity field

Now, we need to capture all influence interactions and resolve them into their respective appreciation and control interactions. Although there are six influence interactions, for the sake of simplicity, we will choose only three.

Table I.18 AIC interactions table for I.8 model

Source or Sink: {avp_development_datasets}	
Output Behaviour	**Input Behaviour that impacts the emergence of Output Behaviour**
I1: |{ avp_development_datasets}_[+assure]_ { architect_predictive_mental_ model_ArcPM}|	**A1:** |{ avp_development_datasets}_[+capture]_ { other_flying_objects}| **A2:** |{ avp_development_datasets}_[+capture]_ { weather_conditions_variety_definition}|
	C1: |{avp_development_datasets}_[?]_{?}|
I2: |{ avp_development_datasets}_[+assure]_ {informed_air_traffic_controller}|	**A3:** |{avp_development_datasets}_[?]_{?}|
	C2: |{avp_development_datasets}_[?]_{?}|
I3: |{ other_flying_objects}_[-visually complicate]_ { avp_development_datasets }|	**A4:** |{other_flying_objects}_[?]_{weather_conditions}|

	C3: |{other_flying_objects}_[?]_{?}|

I.6.2.5 Step 2.5) Capture the AIC interactions

To determine the necessary control interaction (C1), in table I.18, in the context of achieving I1, we need to identify a scenario where the AVP development datasets exert a high likelihood of direct governance over a critical aspect of the AVP system's operation or the environment that the AVP system directly relies on. In AIC terminology, this would mean the datasets have a direct, nearly assured impact on a system component that is crucial to the AVP's functionality.

Given the context, the control interaction (C1) would likely be centred around the critical assurance of the AVP's detection and identification capabilities**.** This would ensure that the AVP system consistently performs accurately and effectively in recognizing objects that could pose risks or inform pilot decision-making.

Suggested Control Interaction (C1)

C1:
|{avp_development_datasets}_[+predict]_{reliable_correct_target_object_recognition_confidence_scores}|

Explanation:

1. avp_development_datasets control the target_object_recognition_confidence _scores directly to ensure that the AVP system operates within defined, precise boundaries that govern object recognition and identification. The correct prediction of this confidence_score would set high-assurance criteria that enable the AVP system to recognise and respond to objects in the environment, directly impacting the AVP's control over the safety and decision-making processes reliably and accurately.
2. The system can meet safety and operational requirements without uncertainty or failure thanks to the control over the accurate and dependable prediction of recognition confidence scores, which guarantees that AVP's performance in identifying and classifying objects stays consistent and accurate.

Resolving the complicatedness of I2:

To achieve the influence interaction:

I2:|{avp_development_datasets}_[+assure]_{informed_air_traffic_controller}|

Where the AVP development datasets influence the air traffic controller's informed situation by ensuring data accuracy and reliability. To do so, specific control and appreciation interactions are required to establish the necessary foundation for this influence. These supportive interactions will ensure that the AVP development datasets effectively impact air traffic controllers through indirect yet robust information governance.

Here's how these interactions would support I2:

Required Control Interaction

Control Interaction (C2):

|{avp_development_datasets}_[+validate]_{critical_data_accuracy}|

Explanation:

1. **avp_development_datasets** must exert direct control over **critical data validation and accuracy mechanisms**. Through this interaction, the AVP system's data is rigorously validated and kept at a high level of accuracy. The datasets can consistently deliver high-quality information to downstream systems (like air traffic control support) by managing these validation procedures.
2. Because air traffic decisions depend on accurate and validated data, this control interaction is essential to achieving the guaranteed and consistent data standards that enable the AVP system to effectively influence air traffic controllers.

Required Appreciation Interactions

To support the assurance influence on the **air traffic controller**'s informed situation, the following appreciation interactions would provide adaptability and contextual awareness to the AVP datasets, allowing them to account for variations in the operating environment:

Appreciation Interaction (A3):

|{avp_development_datasets}_[+capture]_{visual_traffic_density_and_patterns}|

Explanation:

1. This appreciation interaction allows the AVP development datasets to capture information about **traffic density and patterns**, which vary with time, location, and operational context. By adapting to these patterns, the AVP system can better ensure that air traffic controllers are informed about its awareness of up-to-date traffic information (real-time). So, the AVP needs to communicate its awareness of traffic density so that the ATC can verify that AVP is aligned with reality.
2. The ability to capture traffic patterns aids the datasets in offering relevant insights to controllers, aligning with real-world conditions and supporting informed decision-making.

Resolving the complicatedness of I3:

In the context of other_flying_objects visually complicating the avp_development_datasets (interaction I3): |{other_flying_objects}_[−visually complicate]_{avp_development_datasets}|

here is an analysis of the necessary control and appreciation interactions for other_flying_objects to support this influence:

Control Interaction for other_flying_objects (C3)

To create visual complications for the AVP system, other_flying_objects directly govern specific visual parameters that directly impact how the AVP system perceives them. This control interaction would allow other flying objects to directly govern aspects of their appearance or behaviour in ways that affect their visual complexity and the resulting perception challenges for the AVP.

Control Interaction (C3):

|{other_flying_objects}_[+control]_{other_flying_objects _visual_complexity_parameters}|

Explanation:

1. other_flying_objects exert control over visual_complexity_parameters, such as shape, colour variations, speed, and manoeuvre patterns, to increase visual complexity for the AVP system. This control allows them to adjust their appearance and behaviour directly, making it more challenging for the AVP system to recognise, classify, and track them accurately.
2. By managing these parameters, other flying objects create an environment where the AVP system must deal with varying and potentially conflicting visual data, which complicates its recognition processes and overall situational awareness.

Appreciation Interaction for other_flying_objects (A4)

Other flying objects also need to appreciate external environmental conditions, particularly weather conditions, to enhance the effectiveness of their visual complication impact on the AVP system. Weather directly affects visibility, lighting, and sensor effectiveness, influencing how the AVP system perceives other objects.

Appreciation Interaction (A4): |{other_flying_objects}_[manoeuvre]_{weather_conditions}|

Explanation:

1. By appreciating weather conditions, other flying objects have to adjust their behaviours, which amplifies visual complications in specific weather scenarios. For instance, certain movements or colour patterns might make the AVP system's object identification more challenging during foggy or low-visibility conditions.

Given the above evaluation the following is the updated AIC interactions:

Table I.19 Resolved AIC Interactions Table

Source or Sink: {avp_development_datasets}	
Output Behaviour	**Input Behaviour that impacts the emergence of Output Behaviour**
I1: |{ avp_development_datasets}_[+assure]_ { architect_predictive_mental_ model_ArcPM}|	**A1:** |{ avp_development_datasets}_[+capture]_ { other_flying_objects}| **A2:** |{ avp_development_datasets}_[+capture]_ { weather_conditions_variety_definition}|
	C1: |{avp_development_datasets}_[+predict]_{reliable_correct_target_object_recognition_confidence_scores}|
I2: |{ avp_development_datasets}_[+assure]_ {informed_air_traffic_controller}|	**A3:** |{avp_development_datasets}_[+capture]_{visual_traffic_density_and_patterns}|
	C2: |{avp_development_datasets}_[+validate]_{critical_data_accuracy}|
I3: |{ other_flying_objects}_[-visually complicate]_ { avp_development_datasets }|	**A4:** |{other_flying_objects}_[manoeuvre]_{weather_conditions}|

	C3: |{other_flying_objects}_[+control]_{other_flying_objects _visual_complexity_parameters}|

There are a number of new systems have been discovered from the above table. We can capture those new systems by extending on icon definition I.16. Below is the new extended I.16 table:

Table I.20 Extended 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	
reliable_correct_target_object_recognition_confidence_scores	
critical_data_accuracy	
visual_traffic_density_and_patterns	
other_flying_objects _visual_complexity_parameters	

Now we extends the AIC model schema in I.8 to include the newly added complexes. Figure I.9 captures the newly extended complexity field.

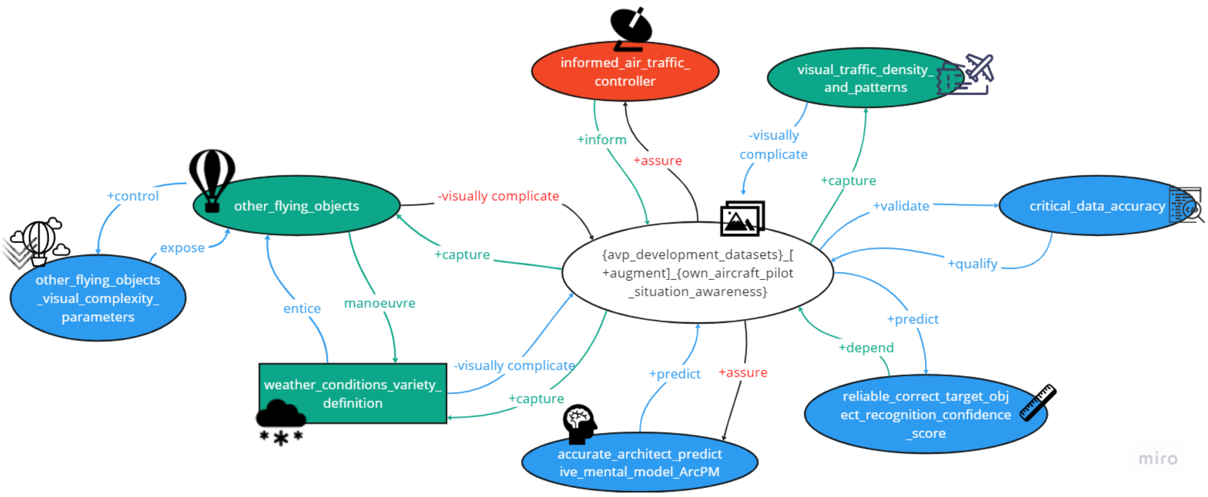


Figure I.9 Extended AIC Schema

Note that the red influence interactions have been changed to black (keeping the action in red). This signifies that those influence interactions have been resolved to their appreciation and control components. The table acts as a description of the complexity dynamics of the complexity field above. The next thing to do is to resolve the new complicatedness comprehensively among the newly discovered factors:

Table I.21 Extended AIC Actions Matrix which includes all hidden Black Swan scenarios

		+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 complicat e	visually complicate	-visually complicat e	-visually complicat e	- operationall y complicate	visually complicate	reduce reliability		visually complicat e
	+qualify	+expose	capture	+assure	+assure	+capture	+predict	+capture	

Given the above new factors, we need to construct a comprehensive complexity field. To do so, we will need to construct the AIC actions matrix to identify all complexes with each other.

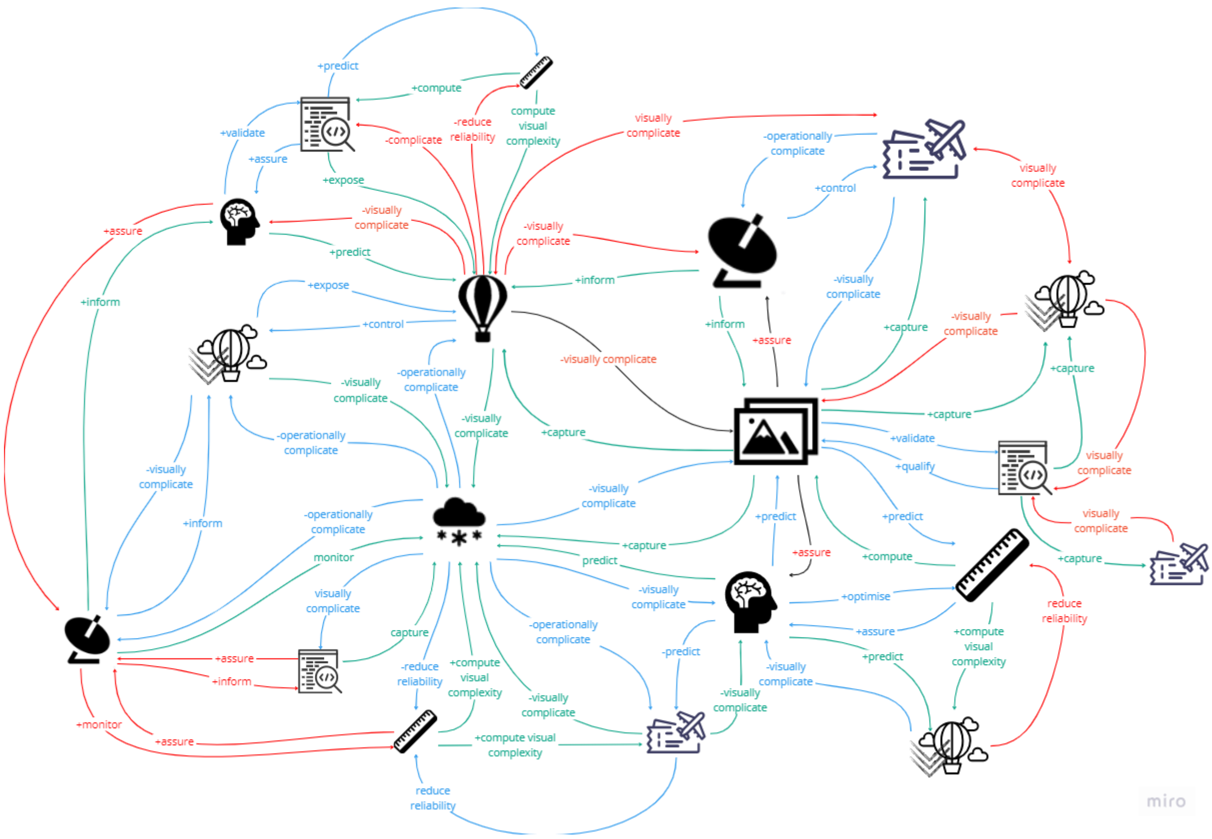


Figure I.10 The latest iteration of AIC Complexity Field for AVP Dataset Development capturing AIC Actions Matrix I.21

I.6.3 Predictive Thinking Pipeline 3: Hazards, Threats and Opportunities Scenarios (HazTOPS) Analysis

We apply the HazTOPS SECoT outlined in section [reference]. The following is the application of the process:

I.6.3.1 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.

We will mainly apply the HazTOPS analysis to I.9 Schema in order to simplify the process.

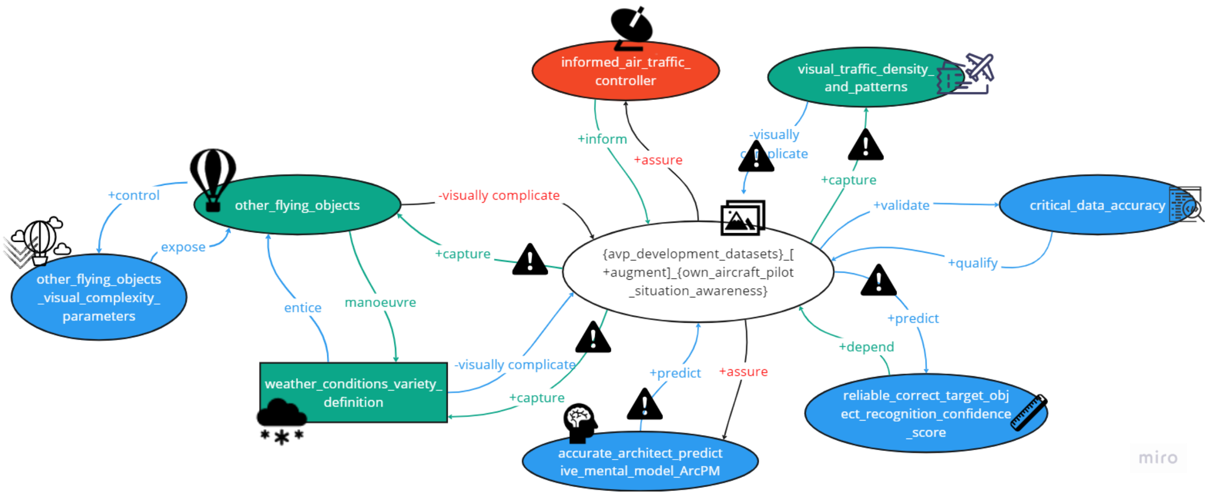


Figure I.11Sources of Hazards AIC Complexity Field

Figure I.11 presents a HazTOPS analysis applied to the AIC Complexity Field for identifying safety hazards, and operational challenges. The diagram visualises the interactions between AVP development datasets, environmental factors, air traffic control systems, and data reliability, highlighting areas where hazards my emerge.

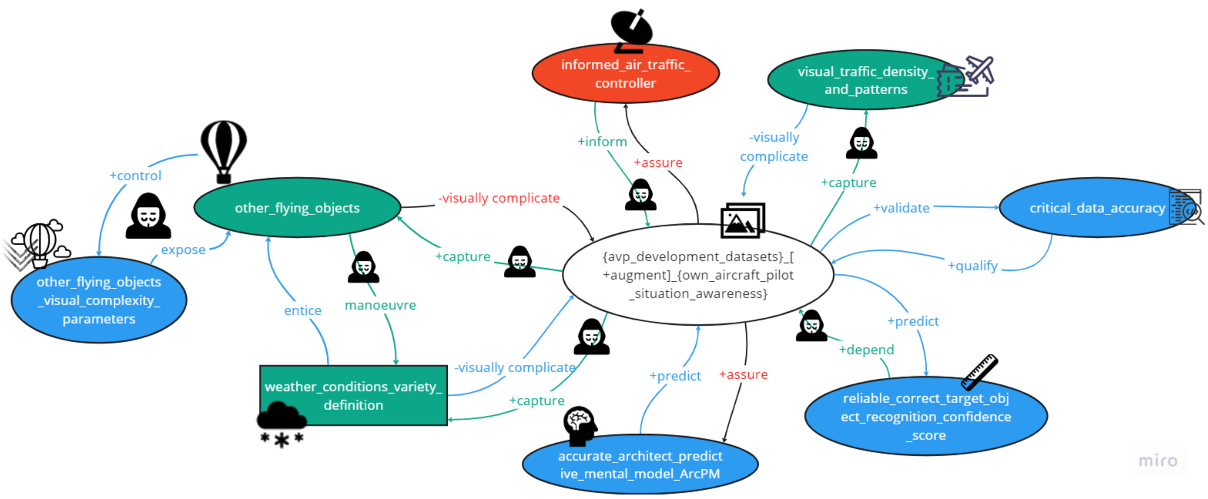


Figure I.12 Sources of Threats AIC Complexity Field

Figure I.12 presents a HazTOPS analysis applied to the AIC Complexity Field for identifying potential threats and vulnerabilities within the operational environment. The diagram visualises the interactions between AVP development datasets, environmental conditions, air traffic control systems, and critical data accuracy, highlighting areas where threats may emerge. Key concerns include visual complexity, misclassification risks, and data reliability issues, which could compromise the architect's situational awareness and AVP dataset development processes. This structured representation aids in assessing systemic threats and enhancing resilience against potential security and operational challenges in complex aviation systems.

I.6.3.2 Step 2) Characterise the scoped interactions.

In this step, we will take only consider the hazards of the complexity field:

Table I.22 Considering Hazards related to I1 interaction

Source or Sink: {avp_development_datasets}	
Output Behaviour	**Input Behaviour that impacts the emergence of Output Behaviour**
I1: |{ avp_development_datasets}_[+assure]_{ architect_predictive_mental_ model_ArcPM}|	**A1:** |{ avp_development_datasets}_[+capture]_{ other_flying_objects}| **A2:** |{ avp_development_datasets}_[+capture]_{ weather_conditions_variety_definition}|
	C1: |{avp_development_datasets}_[+predict]_{reliable_correct_target_object_recognition_confidence_scores}|

I.6.3.3 Step 3) Apply predictive potential complications guide words.

Then, 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.

Table I.23 Example "More" guide word complication

Title	Multiple flying objects (other than aircrafts)
AIC interaction	**A1:** |{ avp_development_datasets}_[+capture]_ { other_flying_objects}|
HazTOPS Aspect	**Definition**
Hazards, Threats or Opportunities Scenario: Guide word	More: multiple balloons involved.

Operating Scenario Context	Clear day, ownship aircraft (with AVP) at 10000ft high.
Hazard, Threat or Opportunity definition (consider used-systems)	More than 1 hot air balloon is visible. At 3000 ft.
Foreseeable Sequence of Events	• Planned a recreational or observational event that involves multiple balloons. • A group of hot air balloons or other balloons (e.g., weather or recreational balloons) are launched in the vicinity of the aircraft's flight path. • The AVP system identifies multiple objects (balloons). • The system classifies these objects based on altitude, size, and motion patterns. • Ownship pilot or automated system receives an alert of potential objects within or near the flight path.
Potential harm or benefit	• The shape of hot air balloons may startle the AVP, leading to a false negative. • The number and location of the hot air balloons may startle the AVP, leading to a false negative.

Table I.23 explores the application of the "More" guide word within a HazTOPS predictive complexity analysis, focusing on the potential impact of multiple flying objects—specifically, the presence of more than one hot air balloon in the operational environment of an ownship aircraft equipped with AVP at 10,000 feet altitude.

The AIC interaction under analysis is:

A1: |{avp_development_datasets}_[+capture]_{other_flying_objects}|

Where the AVP system captures and processes data related to airborne objects in its vicinity. The HazTOPS aspect introduces the "More" guide word, indicating an increase in the number of detected airborne objects—particularly multiple hot air balloons at lower altitudes (3,000 feet).

I.6.4 Predictive Thinking Pipeline 4: Elicitate ordered AIC System-Level Requirements and training requirements

In this step, we will model every derived HazTOPS and define mitigating system-level requirements. We will perform the process in the following 2 examples:

I.6.4.1 Step 1) Model the Complex of Interest Operating Scenario Context

At a high level, we model the AVP as a source and the AIC relationships with the required used-systems to achieve the emergent capability. First, we identify the unresolved influence relationship problem between the agent and the target object of interest in Figure I.13:

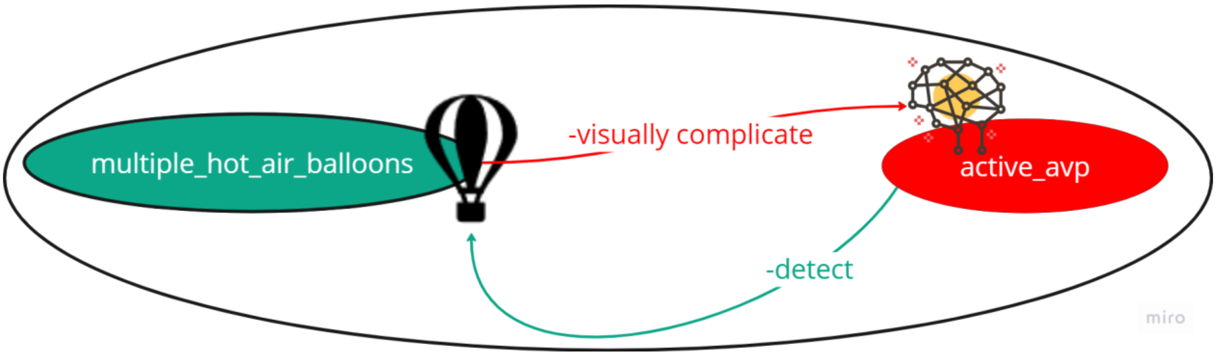


Figure I.13 Initial abstraction of Eagle Robot (ER) Agent emergent capability boundary

The model multiple_hot_air_balloons negatively complicates the perception system of active Aircraft Collision Avoidance Perception systems (AVP). It describes how the presence of hot air balloons can interfere with or degrade the performance of an AVP system, which is designed to detect and help avoid aircraft. Below are a number of unknown unknowns that we need to predict in Table I.24.

Table I.24 Initial AIC resolution interactions table

Agent Output Behaviour	**Agent Input Behaviour that delivers Output Behaviour**	**Supportive systems AIC Behaviour**
I1: { multiple_hot_air_balloons_ }_[-visually complicate]_{active_avp}	**A:** to be defined	None
	C: To be defined	
No influence from AVP but a higher level of appreciative interaction **A4:** { active_avp}_[-detect]_{multiple_hot_air_balloons}	**A:** { active_avp}_[detect]_{ multiple_hot_air_balloons}	
	C: To be defined	

I.6.4.2 Step 2) Model hazards mitigation Ordered-AIC complexity field

We must resolve the influence relationship problem on its appreciation and control relationships. To refine this model into subsystems, we can break down the visual components that define the appearance of hot air balloons as seen by the AVP system and the components of the ML-based perception model used to interpret these visuals. This will help understand how the AVP system could handle the visual complexity introduced by multiple balloons and improve detection accuracy.

Visual Components of Hot Air Balloons (as Perceived by AVP)

These are the key **visual attributes** that the AVP system might use to detect and differentiate hot air balloons from other objects:

- **Colour and Pattern**: Hot air balloons are often brightly coloured with unique patterns, making colour segmentation a useful feature. The perception system may analyse colour contrasts to distinguish balloons from the background and each other.
- **Shape and Contour**:
  - **Silhouette**: Hot air balloons have a distinctive, often rounded shape with a narrower base where the basket is located.
  - **Edge**: The balloon and basket structure has unique edges that can be detected even from a distance. The AVP system might use contour-based filters to recognise these shapes.
- **Size and Scale**: The balloon's apparent size changes with distance, which the AVP can use to estimate proximity. This may involve analysing relative scale based on prior balloon size data.
- **Texture and Surface Features**: Balloons have smooth surfaces, sometimes with logos or text, which may require texture-based recognition methods to enhance detection, especially if the AVP system is differentiating between types of balloons.

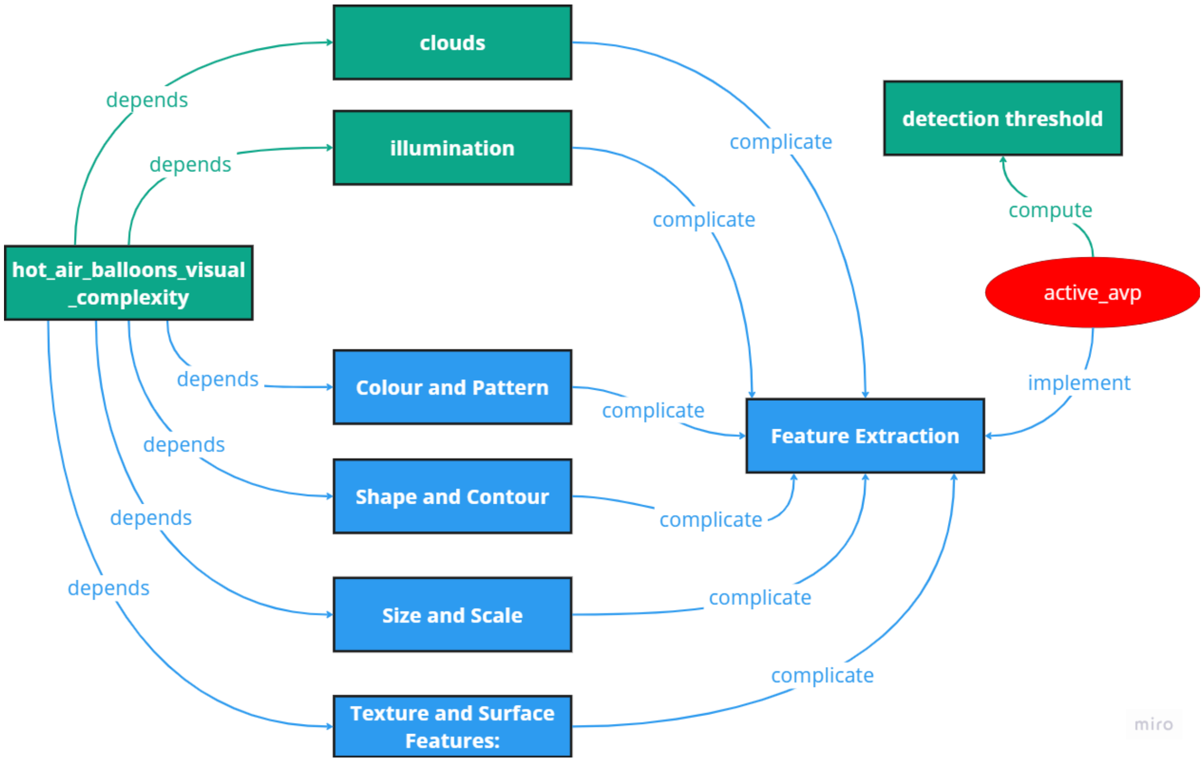


Figure I.14 Resolution of multiple hot air balloons hazard

The following table captures the description of the Figure I.14:

Table I.25 AIC-Structured Interactions Detailing the Visual Complexity of Hot Air Balloons and Its Impact on AVP Perception

Agent Output Behaviour	**Agent Input Behaviour that delivers Output Behaviour**	**Supportive systems AIC Behaviour**
I1: { hot_air_balloons_visual_complexity}_[-visually complicate]_{active_avp}	**A1:** { hot_air_balloons_visual_complexity}_ [depends]_{clouds} **A2:** { hot_air_balloons_visual_complexity}_ [depends]_{illumination}	None
	C1: { hot_air_balloons_visual_complexity}_ [depends]_{ Colour and Pattern} **C2:** { hot_air_balloons_visual_complexity}_ [depends]_{ Shape and Contour}	

	C3: { hot_air_balloons_visual_complexity}_ [depends]_{ Size and Scale} **C4:** { hot_air_balloons_visual_complexity}_ [depends]_{ Texture and Surface Features}	
No influence from AVP but a higher level of appreciative interaction **A4:** { active_avp}_[-detect]_{multiple_hot_air_balloons}	**A3:** { active_avp}_[compute]_{ detection confidence_score}	
	C5: { active_avp }_[implement]_{ Feature Extraction}	

Table I.25 presents an AIC-structured analysis of how hot air balloons' visual complexity influences the Active AVP system's perception and detection capabilities. The table categorises interactions based on AIC functions, mapping hot air balloon attributes that complicate AVP visual processing and the supportive systems required for perception enhancement.

The influence interaction (I1) highlights the negative visual complications introduced by hot air balloons, affecting the AVP's ability to classify and track airborne objects. The appreciation interactions (A1-A2) establish dependencies on environmental factors such as cloud cover and illumination, further impacting object visibility and detection reliability. The control interactions (C1-C4) focus on the specific visual attributes of hot air balloons, including colour, pattern, shape, contour, size, scale, texture, and surface features, all contributing to perceptual uncertainty for AVP systems.

The table also recognises that AVP does not inherently influence hot air balloons, meaning it must adapt its perception strategies instead of modifying external conditions. The AVP compensates by computing detection confidence scores (A3) and implementing feature extraction techniques (C5) to enhance its ability to distinguish hot air balloons from other airborne objects.

I.6.4.3 Step 3) Ordered-AIC-based Mitigating System or Safety Requirements Derivation (Safe Operating Concept)

So far, we have focused on defining Agent relationships in the given hazardous scenario. In this step, we collate all discovered actions in a table and convert those actions into AVP systems requirements, considering the intangible or tangible methods that shall deliver the agent's will. The relevant system requirements are defined from Control and Appreciation actions, with the influence action included at the end of each requirement after the phrase "in order to". Use the following format:

[**Given:** A or C actions] **in order to** [I action], **Then** [mitigation requirement] **In order to** [I action]

For example:

AVP Safety Requirements:

Here are the AVP System safety requirements based on the given actions (A and C actions) for mitigating the complexity introduced by multiple hot air balloons and enhancing AVP's ability to detect and classify them reliably:

Table I.26A Safety requirements derivations to mitigate hot-air balloons impact on AVP system performance.

AC (appreciative, control) interaction	Mitigating Safety or Systems Requirements (Safe Operating Concept)
A1: { hot_air_balloons_visual_complexity}_ [depends]_{clouds}	**Safety Requirement 1: AVP System shall avoid Hot Air Balloons among various cloud conditions** • **Given**: A1: Hot air balloon visual complexity depends on cloud cover, • **In order to**: visually complicate AVP System performance (I1), • **Then**: The AVP system shall guide ownship aircraft to avoid hot air balloons under various cloud cover scenarios **In order to**: recognise and direct ownship aircraft to avoid hot-air balloon visual complexity.

A2: { hot_air_balloons_visual_complexity}_ [depends]_{**illumination**}	**Safety Requirement 2: AVP system to avoid Hot Air balloons in various light conditions** • **Given**: A2: Hot air balloon visual complexity depends on illumination conditions, • **In order to**: visually complicate AVP System performance (I1), • **Then**: The AVP system shall guide ownship aircraft to avoid hot air balloons under various illumination conditions, • **In order to**: recognise and direct ownship aircraft to avoid hot-air balloon visual complexity.
C1: { hot_air_balloons_visual_complexity}_ [depends]_{ Colour and Pattern}	**Safety Requirement 3: AVP System shall avoid Air balloons with rare patterns** • **Given**: C1: Hot air balloon visual complexity depends on colour and pattern variations, • **In order to**: visually complicate AVP System performance (I1), • **Then**: The AVP system shall guide ownship aircraft to avoid a broad range of balloon colours and patterns, including common and rare patterns and colour schemes, • **In order to**: recognise and direct ownship aircraft to avoid hot-air balloon visual complexity.
C2: { hot_air_balloons_visual_complexity}_ [depends]_{ Shape and Contour}	**Safety Requirement 4: AVP System shall avoid Hot Air balloons with rare shapes and contours** • **Given**: C2: Hot air balloon visual complexity depends on shape and contour, • **In order to**: visually complicate AVP System performance (I1),

	• **Then**: The AVP system shall guide ownship aircraft to avoid varied balloon shapes and contours, including partially obscured or overlapping balloons, as well as balloons viewed from multiple angles, • **In order to**: recognise and direct ownship aircraft to avoid hot-air balloon visual complexity.
C3: { hot_air_balloons_visual_complexity}_ [depends]_{ Size and Scale}	**Safety Requirement 5: AVP System shall avoid Hot Air balloons at various distances** • **Given**: C3: Hot air balloon visual complexity depends on size and scale, • **In order to**: visually complicate AVP System performance (I1), • **Then**: The AVP system shall guide ownship aircraft to avoid balloons at various distances and sizes, emulating both close-range and far-range balloon appearances, • **In order to**: recognise and direct ownship aircraft to avoid hot-air balloon visual complexity.
C4: { hot_air_balloons_visual_complexity}_ [depends]_{ Texture and Surface Features}	**Safety Requirement 6: AVP system shall avoid Hot Air balloons with varying textures** • **Given**: C4: Hot air balloon visual complexity depends on texture and surface features, • **In order to**: visually complicate AVP System performance (I1), • **Then**: The AVP system shall guide ownship aircraft to avoid balloon images with varying textures, surface features, and common markings.

	In order to: allow the AVP system to differentiate balloons based on subtle texture and surface features, such as logos or text.
A3: { active_avp}_[compute]_ { detection confidence_score}	**System Requirement 1: The AVP system shall adjust confidence thresholds dynamically** **Given**: A3: Active AVP computes detection confidence score, **In order to**: enable AVP to perform detection under complex visual conditions (A4), **Then**: The AVP system shall implement a detection confidence scoring mechanism, adjusting thresholds based on scenario-specific complexities such as high object density and overlapping visuals. **In order to**: enable the AVP to flag uncertain detections for further processing or alerting when confidence is low, enhancing overall safety (I1).
C5: { active_avp }_[implement]_ { Feature Extraction}	**System Requirement 2: The AVP system shall implement feature extraction** **Given**: C5: Active AVP implements feature extraction, **In order to**: enable AVP to perform detection under complex visual conditions (A4), **Then**: The AVP system shall use feature extraction techniques, including attention mechanisms to prioritise relevant balloon features, **In order to**: detect hot air balloons.

By methodically taking into account how the AVP system should manage hot air balloons under various visual complexities, the safety requirements derived from the Appreciative and Control (AC) interactions in Table I.26A were improved.

Appreciative actions and safety requirements:

The appreciative actions (A1, A2, A3) highlight the ways in which the detectability of the visual complexity of hot-air balloons is influenced by external factors like cloud cover, illumination, and system confidence computation. These components impact AVP's performance and change the visual scene.

- **A1: Dependence on Cloud Cover/ Safety Requirement 1:** False positives or missed detections may result from hot air balloons visually blending with clouds. The AVP system needs to make sure it can detect hot-air balloons in a variety of cloud conditions because cloud cover changes contrast. Ownship aircraft must be guided by the system to steer clear of situations where clouds obscure balloon visibility.
- **A2: Dependence on Illumination/ Safety Requirement 2:** The AVP system's perception of objects is impacted by variations in illumination. In order to avoid misclassifying hot-air balloons, this requirement ensures that the AVP system can adjust to various lighting conditions, such as glare, low light, and shadowed environments.
- **A3: Computation of Detection Confidence/System Requirement 1:** To handle ambiguous visual conditions, the AVP system needs to dynamically modify detection confidence scores. Under certain operating conditions (such as when visibility decreases or clouds become overcast), this requirement introduces an adaptive scoring mechanism. To guarantee the most likely detection, the system should modify its confidence thresholds. On the basis of overlapping or unclear features, the AVP can flag uncertain detections. This could prevent aircraft manoeuvres that are missed or inaccurate because of faulty detections.

Control actions and system adaptations:

The control actions (C1–C5) are related to the AVP system's capability to actively distinguish and mitigate balloon visual complexity through shape, colour, size, and other object features.

- **C1: Dependence on Colour and Pattern / Safety Requirement 3:** It can be difficult to identify hot-air balloons due to their diverse and erratic colours and patterns. In order to avoid confusion with aircraft, background scenery, or other environmental objects, the requirement guarantees that the AVP system can handle balloons with both common and uncommon designs.
- **C2: Dependence on Shape and Contour / Safety Requirement 4:** The AVP system's perception of objects is affected by the shape of the balloons (such as spherical, elongated, or specially themed balloons). Certain shapes might blend into the

background or seem partially obscured by other objects. To avoid misclassification, the requirement makes sure the AVP system can identify various balloon contours from a variety of viewing angles.

- **C3: Dependence on Size and Scale / Safety Requirement 5:** The relative scale and distance of a hot-air balloon can lead to detection ambiguity. At different heights, balloons can appear as big, close-by obstacles or as tiny, far-off objects. To ensure that avoidance manoeuvres are carried out correctly, the AVP system must be able to distinguish balloons at various scales.
- **C4: Dependence on Texture and Surface Features / Safety Requirement 6:** Text, logos, and uneven textures on hot-air balloons can interfere with object detection. This requirement ensures that the AVP system can account for subtle texture differences to improve detection accuracy and avoid misclassification or missed detections.
- **C5: Implementation of Feature Extraction / System Requirement 2:** The AVP system must implement advanced feature extraction mechanisms to enhance recognition, using attention-based models to prioritise balloon-related features. This requirement ensures that the AVP system can filter out irrelevant background noise and focus on extracting balloon-specific characteristics.

I.6.4.4 Step 4) Extended Concrete Safety, Systems requirements and ML Safety-Training Concept

We will not apply the 4-whats-and-how method in this case study since we have already derived the AVP safety requirement. We will directly derive the training concept. Instead, we will move to the derivation of the training concept.

I.6.4.5 ML Safety-Training Requirement derivation (Training Concept):

In this stage, we specify a general training requirement for the ML model, which we refer to as the Training Concept. To define the training requirement, we used the following structure:

ML Safety-Training Requirement 1: [system of interest] shall be trained to [training experience].

Below is the table we use to perform the derivation process:

Table I.26B Deriving Training Concept from Safe Operating Concept for AVP ML component

Mitigating Safety or Systems Requirements (Safe Operating Concept)	**ML Safety-Training Requirements (Training Concept)**
Safety Requirement 1: AVP System shall avoid Hot Air Balloons among various cloud conditions • **Given**: A1: Hot air balloon visual complexity depends on cloud cover, • **In order to**: visually complicate AVP System performance (I1), • **Then**: The AVP system shall guide ownship aircraft to avoid hot air balloons under various cloud cover scenarios. • **In order to**: recognise and direct ownship aircraft to avoid hot-air balloon visual complexity.	**ML Safety-Training Requirement 1**: AVP ML component shall be trained to recognise Hot Air Balloons under various cloud cover scenarios (e.g., clear, partial cloud cover, heavy cloud cover),
Safety Requirement 2: AVP system to avoid Hot Air balloons in various light conditions • **Given**: A2: Hot air balloon visual complexity depends on illumination conditions, • **In order to**: visually complicate AVP System performance (I1), • **Then**: The AVP system shall guide ownship aircraft to avoid hot air balloons under various illumination conditions, • **In order to**: recognise and direct ownship aircraft to avoid hot-air balloon visual complexity.	**ML Safety-Training Requirement 2:** AVP ML component shall be trained to recognise a Hot Air Balloon under various illumination conditions (e.g., dawn, dusk, direct sunlight, low-light),

Safety Requirement 3: AVP System shall avoid Air balloons with rare patterns • **Given**: C1: Hot air balloon visual complexity depends on colour and pattern variations, • **In order to**: visually complicate AVP System performance (I1), • **Then**: The AVP dataset shall include a broad range of balloon colours and patterns, including common and rare patterns and colour schemes, **In order to**: recognise and direct ownship aircraft to avoid hot-air balloon visual complexity.	**ML Safety-Training Requirement 3:** AVP ML component shall be trained to recognise a broad range of balloon colours and patterns, including common and rare patterns and colour schemes.
Safety Requirement 4: AVP System shall avoid Hot Air balloons with rare shapes and contours • **Given**: C2: Hot air balloon visual complexity depends on shape and contour, • **In order to**: visually complicate AVP System performance (I1), • **Then**: The AVP dataset shall include varied balloon shapes and contours, including partially obscured or overlapping balloons, as well as balloons viewed from multiple angles, **In order to**: recognise and direct ownship aircraft to avoid hot-air balloon visual complexity.	**ML Safety-Training Requirement 4:** The AVP ML component shall be trained to recognise a Hot Air Balloon in the context of varied balloon shapes and contours, including partially obscured or overlapping balloons, as well as balloons viewed from multiple angles.

Safety Requirement 5: AVP System shall avoid Hot Air balloons at various distances • **Given**: C3: Hot air balloon visual complexity depends on size and scale, • **In order to**: visually complicate AVP System performance (I1), • **Then**: The AVP dataset shall include balloons at various distances and sizes, emulating both close-range and far-range balloon appearances, **In order to**: recognise and direct ownship aircraft to avoid hot-air balloon visual complexity.	**ML Safety-Training Requirement 5:** The AVP ML component shall be trained to recognise Hot Air balloons at various distances and sizes, emulating both close-range and far-range balloon appearances.
Safety Requirement 6: AVP system shall avoid Hot Air balloons with varying textures • **Given**: C4: Hot air balloon visual complexity depends on texture and surface features, • **In order to**: visually complicate AVP System performance (I1), • **Then**: The AVP dataset shall contain balloon images with varying textures, surface features, and common markings. **In order to**: allow the AVP system to differentiate balloons based on subtle texture and surface features, such as logos or text.	**ML Safety-Training Requirement 6:** The AVP ML component shall be trained to recognise a Hot Air Balloon in the context of varying textures, surface features, and common markings.
System Requirement 1: The AVP system shall adjust confidence thresholds dynamically	**ML Training Requirement 1 (non-safety):** The AVP ML component shall be trained to adjust confidence thresholds dynamically and perform under various confidence scoring mechanisms.

• **Given**: A3: Active AVP computes detection confidence score, • **In order to**: enable AVP to perform detection under complex visual conditions (A), • **Then**: The AVP system shall implement a detection confidence scoring mechanism, adjusting thresholds based on scenario-specific complexities such as high object density and overlapping visuals. • **In order to**: enable the AVP to flag uncertain detections for further processing or alerting when confidence is low, enhancing overall safety (I1).	The thresholds shall be adjusted based on varying environmental conditions, high TOI density and pictorial distances.
System Requirement 2: The AVP system shall implement feature extraction • **Given**: C5: Active AVP implements feature extraction, • **In order to**: enable AVP to perform detection under complex visual conditions (A4), • **Then**: The AVP system shall use feature extraction techniques, including attention mechanisms to prioritise relevant balloon features, • **In order to**: detect hot air balloons.	Not applicable

I.6.4.6 ML dataset requirements derivation

A dataset training requirement can be derived from the training concept. We use the following structure to define requirements over actual datasets:

Dataset requirement structure:

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

In this case:

> **ML Development Dataset Requirement 1:** The AVP's ML component Training Dataset shall provide the trainee model with a valuable minimum variety of various types of hot air balloons under various cloud cover scenarios (e.g., clear, partial cloud cover, heavy cloud cover).

I.7 Stage 3B: Comprehensive Operational Environment Definition[4]

We reduced the comprehensive ODD classification into the following categories for this application and considered Grade A uncertainty. For a detailed description of how to define ODD, refer to sections E. 9 and E. 10.

Table I.27 Operational Design Definition for AVP

System of interest	Computerised perception-based mid-air collision avoidance system (AVP)
Solution Operational Space	Open Airspace [Palo Alto City, USA]
ODD Uncertainty Grade	Grade A
Development purpose	Training and Testing
Environment Characteristics	**Ideal natural environment** system for perception
Natural Lighting Conditions	Sunny

[4] For a summary see Section M.5

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

Table I.30 defines the operational design domain (ODD) and environmental characteristics for deploying the AVP system in Palo Alto city open airspace under Grade A uncertainty. It specifies ideal natural conditions for perception, including sunny weather, minimal wind (<5 km/h), high visibility (>8 km), and clear skies. Additional parameters include a temperature range of 15–25°C, 12 hours of sunshine, and no pollen, sand, or dust interference. The operation is limited to a single type of season, midday timing, and one specific landscape type with minimal infrastructure variation. The table ensures a clearly defined, controlled environment for reliable system performance and accurate perception during deployment. We may choose different sets for ODD when developing testing datasets. However, we will keep the same ODD for generating the datasets in this application.

I.8 Stage 4: Disordered AIC-Driven Black Swan Scenarios Prediction[5]

In this stage, you may consider disordered AIC timing in section E4. Some of the output of this stage is to generate Black Swan validation datasets for validating ML components to handle Black Swan scenarios. However, nothing may stop the architect from dedicating some of the Black Swan scenarios to be part of the training and testing process of the ML component.

At first, we choose the complexity field we intend to perform the AIC perspective shift. For this we will select the last complexity field in Figure I.10:

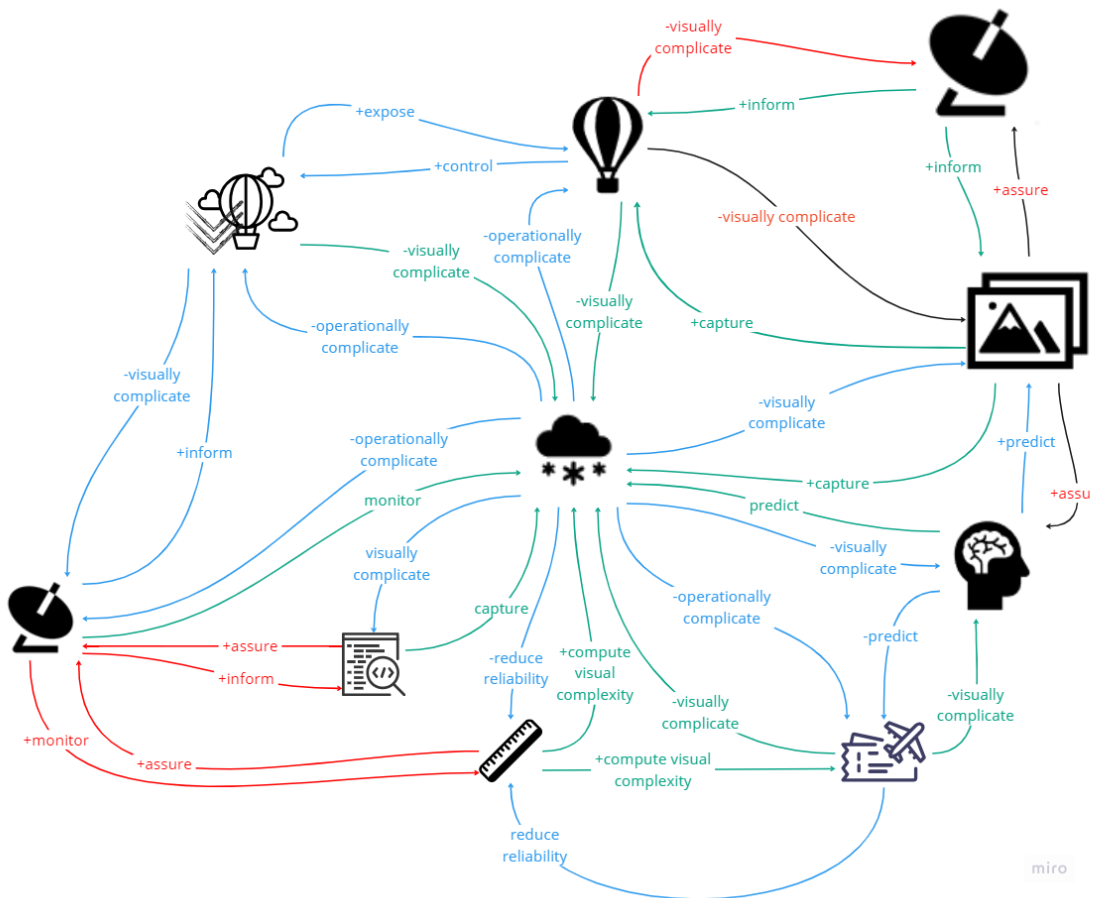


Figure I.15 Chosen complexity field to conduct the AIC perspective shift for Black Swan scenarios discovery

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

[5] For a summary see Section M.6

I.8.1 Step 1) Define the interactions

Define the interactions needed to predict a potential emergence. We chose the following relationship. For this example, we will choose the interaction between other_flying_objects and avp_development_datasets:

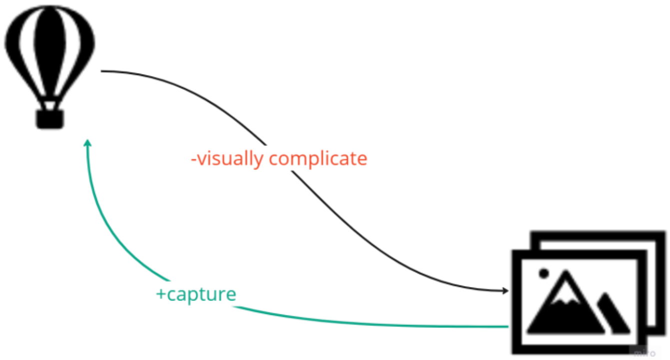


Figure I.16 Chosen interaction (black line means that influence interaction had been resolved into it AC components)

The expected interaction between the hot-air balloon and the AVP dataset development is that the dataset should respond to the balloon's behaviour. The dataset should never influence the balloon's behaviour, regardless of how the complexity of the future field may evolve. The datasets are claimed to be supportive of balloons' purpose (based on the assumption that recognising hot-air balloons and including them in the datasets would contribute to their safety). At the same time, the unexpected variability of hot-air balloons would obstruct the reliability of AVP perception. This is what the architect is prophesising: what would happen in such a scenario. Any deviation from this established perspective norm would be considered unexpected or surprising, characteristics typical of Black Swan events.

I.8.2 Step 2) Define the ArcMatrix

Define the interaction's current AIC factors using the ArcMatrix (section F.2.1). We will give the hot air balloon a descriptive situation of “visually_variable_hot_air_balloon”.

Table I.28 Deep AIC factorisation for I.16 interaction

	Visually_variable_hot_air_objects	avp_development_datasets
Visually_variable_hot_air_objects		**Supra Source:** other_flying_objects.

		PrimeP: perform uninterrupted missions. **Goal:** avoid any obstacle to mission execution. **Goal type:** Influence. **Action:** visually complicate AVP datasets. **Action type: Infleuce**. **Effect:** Obstructive.
avp_development_datasets	**Supra Source:** Ownship aircraft. **PrimeP:** Maintain safe operational flight and avoid mid-air collisions. **Goal:** complete coverage of potential operational scenarios. **Goal type:** Appreciation**.** **Action:** capture scenarios. **Action type:** appreciation. **Effect:** Obstructive.	

I.8.3 Step 3) Perform the Perspective Shift

Perform the perspective shift using AIC perspective shift SECoT. In this step, we choose an appropriate perspective shift that we believe will be a potential black-swan event not foreseen during modelling the interaction between adv_drone_shapes and flying_eagle_drone. We will select the following shift:

1. **AIC type shift.**

For this, we will apply the following Thought Step from SECoT_3:

General Systems 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.

Completion criteria: The step is considered complete when a scenario demonstrating an alternative AIC goal is detailed.

In this thought process, we will shift the AIC-type perspective as follows As per the following schema:

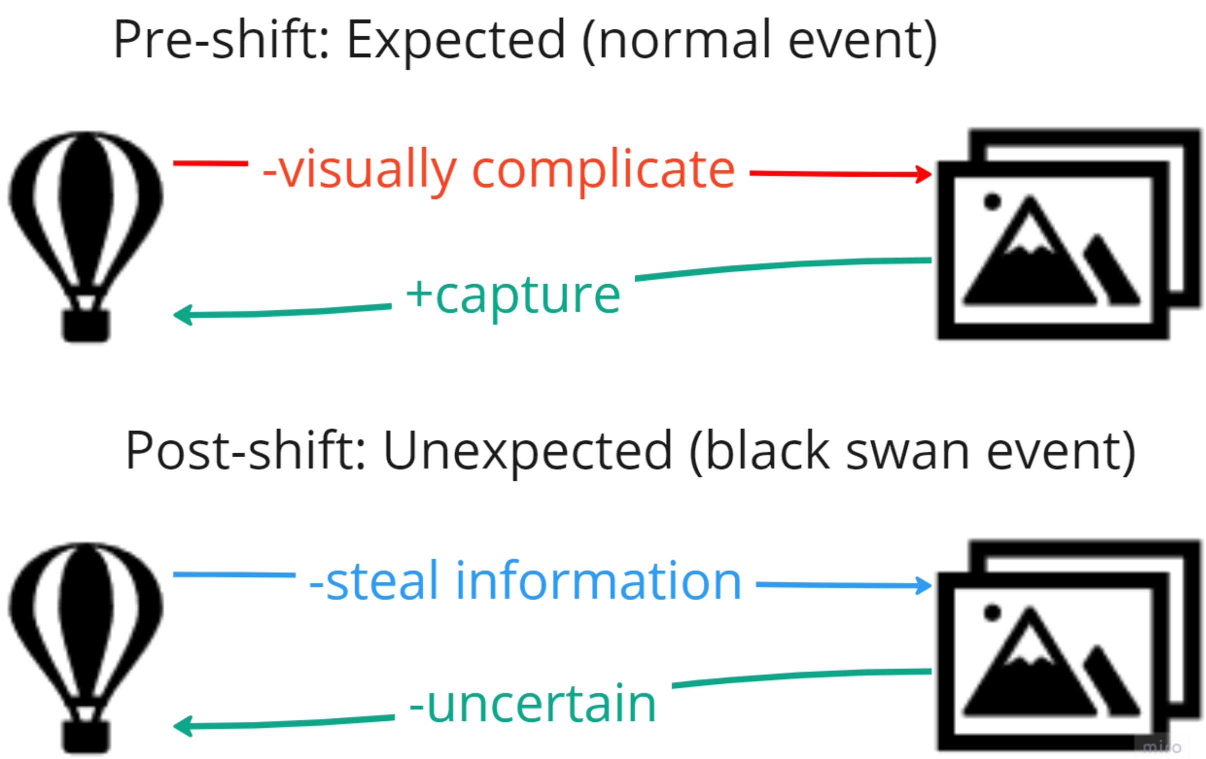


Figure I.17 Shifted perspective from influence to control

The above steps can be captured in the following table:

Table I.29 AIC-type perspective shift

Interaction	**A1:** |{ avp_development_datasets}_[+capture]_ { other_flying_objects}| **I3:** |{ other_flying_objects}_[-visually complicate]_

	{ avp_development_datasets }|	
AIC factors	**Pre-shift perspective**	**Post-shift Perspective**
Source	Visually_variable_hot_air_objects	Visually_variable_hot_air_objects
Sink	avp_development_datasets	avp_development_datasets
Supra Source	other_flying_objects	other_flying_objects
PrimeP	perform uninterrupted missions	perform uninterrupted missions
Source's Goal	**avoid any obstacle to mission execution**	**Spoof the AVP system by using spoofing visual patterns on the hot-air balloon**
Source's Goal type	**Influence**	**Control**
Source's Action	**Visually complicate**	**Steal information of the type of data used for the AVP system**
Source's Action type	**Influence**	**Control**
Source action effect on sink	Obstructive	Obstructive

How do we implement the SECoT_3 thought step above in this context?

The perspective shift captured in Table I.29 was derived using the AIC Type Shift Process, a structured methodology under SECoT_3 that anticipates how complex interactions evolve over time, leading to new situations of complicatedness. The process involves systematically analysing how source-sink relationships change, particularly in how they influence or control their interactions within an operational environment. The steps used to derive the changes in Table I.29 are explained below.

1. Applying AIC Type Shift (General Systems Rule)

The General Systems Rule situations that the goals of a source or sink can change over time due to increased complexity. In this case, the initial interaction was:

A1: |{ avp_development_datasets}_[+capture]_{ other_flying_objects}|

Here, the AVP dataset captures and processes information from other flying objects, including visually variable hot air balloons. The pre-shift perspective assumes that hot air balloons only introduce passive visual complexity into AVP systems (influence-type interaction).

However, by applying the AIC Type Shift, the interaction was reconsidered under future adversarial conditions, leading to an alternative scenario:

I3: |{ other_flying_objects}_[-visually complicate]_{ avp_development_datasets }|

Now, hot air balloons are no longer just visually complex objects; instead, they intentionally spoof the AVP system, shifting their role from passive complexity to an active adversarial control-type goal.

2. Predictive Question Application

The predictive question guides the shift:

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?

Pre-Shift Perspective (Influence)

- The source (visually variable hot air objects) had the goal to avoid obstacles to mission execution, passively complicating the AVP dataset's processing.
- The action type was influence, meaning hot air balloons passively interfered with AVP perception without intent to deceive.
- The effect on the sink (AVP development datasets) was obstructive but unintended.

Post-Shift Perspective (Control)

- The source's goal changed from "avoiding obstacles" to "spoofing the AVP system."
- The source now actively manipulates visual complexity, deploying intentionally deceptive visual patterns to interfere with AVP decision-making.
- The source's action type shifted from influence to control, meaning that hot air balloons are no longer passive obstructions but actively disrupt AVP training and operation.

3. Guiding Prompt Application: Redefining the Nature of Goal and Action

To fully apply the perspective shift, the AIC interactions were re-examined with the following considerations:

- Reviewing AIC Dynamics:
    - Instead of random visual complexity, hot air balloons are now intentionally deceptive, tricking AVP into capturing misleading data.
    - The AVP system, in response, may begin learning incorrect features, weakening its performance.
- Altering Goal and Action:
    - The goal of the balloons changes from passively existing in the environment to actively misleading AVP models.
    - The action shifts from "visually complicate" to "steal information," implying an intelligence-gathering function against AVP datasets.
- Scenario in the Shifted Context:
    - In this new adversarial scenario, hot air balloons are designed with specific patterns to confuse AVP recognition algorithms.
    - AVP dataset integrity is compromised because it learns from misleading visual information, reducing its effectiveness in recognising actual airborne threats.

I.8.4 Step 4) Predict Harder-to-foresee emergent scenarios (Black Swan scenario)

In this step, we elaborate on the scenario further to define the sequence of potential actions. The architect may want to use 5HnWs, or they may use the 5-whys analysis for deeper analysis. To predict Black Swan Scenarios, use the AIC perspective shift as clues and imagine situations where such a scenario may arise. Table I.30 captures the Black Swan scenario.

Table I.30 Harder-to-foresee Emergent Scenario (Black Swan scenario)

Black Swan Scenario	**Rationale for prediction: it is possible that ...**
Black Swan 1: In this **unforeseen scenario**, visually variable hot-air balloons appear in regulated airspace equipped with intentionally **complex visual patterns** that are difficult for the AVP system to classify accurately. These balloons deploy different visual spoofing strategies, such as:	**Impact on AVP System Resilience**: The AVP system, trained in an ordered-AIC environment without these spoofing scenarios, lacks resilience against such complex, variable stimuli. When faced with visually complex patterns intended to exploit gaps in its training data, the AVP system's collision avoidance capability is degraded.

• **Depicting bird-like images or shapes** that are not traditionally detected as hazards within the AVP datasets. • **Displaying visual cues resembling known objects** (e.g., the shape of a common drone or aircraft) but embedded with confusing visual elements, such as unexpected colour schemes or reflective surfaces that alter appearance based on lighting and angle. The hot-air balloons' purpose is to **confuse and mislead the AVP system's object detection and classification algorithms** through repeated, intentional disruption of the AVP's vision processing model. This confusion introduces **obstruction to the AVP's primary purpose**, potentially causing the AVP system to either dismiss real hazards or mistake non-hazardous objects as threats.	**Resulting Vulnerabilities**: This scenario reveals a critical vulnerability: the AVP system's dependency on highly structured datasets makes it susceptible to exploitation by novel or unforeseen patterns, challenging its adaptability and decision-making reliability.

Table I.30 presents a Black Swan scenario that explores an unforeseen adversarial manipulation of the AVP system's object detection and classification algorithms. According to the scenario, visually variable hot air balloons with intricate visual patterns are purposefully created to confuse the AVP system and appear in regulated airspace. Visual spoofing techniques like bird-like imagery, misleading object shapes, and reflective surfaces that change their appearance in different lighting situations are all used by these balloons. This tactic takes advantage of the AVP's reliance on structured training datasets, which could result in misclassification errors and cause the AVP system to ignore real hazards or mistakenly identify non-threats as threats.

I.8.5 Step 5) Define mitigating ML Development and Safety Requirements.

Each safety requirement outlined for Black Swan scenarios necessitates corresponding ML component safety requirements to ensure the AVP can handle unforeseen threats. Below, we define the ML component safety requirements in alignment with each system safety requirement.

Table I.31a AVP Training Requirements for Black Swan Scenarios

Black Swan Scenario	**Safety or systems requirements (Safe Operating Concept)**	**ML Safety-Training Requirement (Training Concept)**
Black Swan 1	**Safety Requirement 2: Enhanced Dataset Diversity** **Given:** The AVP system may misinterpret or fail to detect spoofed objects in its environment. Then: Include in the AVP training dataset a diverse set of complex and deceptive visual patterns (e.g., various colours, lighting angles, reflective surfaces) resembling bird-like objects and drones in controlled airspace. **In order to:** increase the AVP system's resilience against misidentifying spoofed objects and reduce the likelihood of collision risks.	**ML Safety-Training Requirement 2:** The AVP ML component shall be trained to detect, classify, and track diverse set of complex and deceptive visual patterns (e.g., various colours, lighting angles, reflective surfaces) resembling bird-like objects and drones in controlled airspace.
Black Swan 1:	**Safety Requirement 3: Dynamic Perception Algorithms.** **Given:** Standard object classification approaches may fail to recognise atypical or complex spoofing strategies in real time, **Then:** implement adaptive anomaly detection algorithms that detect deviations from typical visual signatures. **In order to:** improve the AVP's ability to recognise unknown or spoofed objects and prompt further verification.	**ML Safety-Training Requirement 3:** The AVP ML component shall be trained to detect deviations from typical visual signatures.

Black Swan 1:	**Safety Requirement 4: Real-Time Adaptive Validation** **Given:** The AVP system may be susceptible to spoofing by visually complex objects, **Then:** integrate real-time cross-checks with air traffic control (ATC) data or other live data sources when confidence is low. **In order to:** validate detected objects against live external information for more accurate decision-making.	**No training required**
Black Swan 1:	**Safety Requirement 4: Security Measures to Safeguard AVP Datasets [non-functional]** **Given:** The AVP datasets may be vulnerable to spoofed data exposure, **Then:** limit access to AVP development datasets and implement strict security protocols for dataset updates. **In order to:** reduce the risk of unauthorised access to AVP data and ensure data integrity.	**No training required**

Black Swan 1:	**Safety Requirement 5: Security Measures to Safeguard AVP Datasets [non-functional]** **Given:** The AVP's data processing could be compromised by intentional data corruption, **Then:** conduct regular security audits and integrity checks on AVP datasets and systems. **In order to:** identify any unauthorised changes to data or system parameters that could impact perception accuracy.	**No training required**

In Table I.31, we structured our process for defining safety and machine learning (ML) training requirements derived from the Black Swan 1 scenario. This situation demonstrates the danger of visually variable hot air balloons that are fitted with misleading patterns that are purposefully meant to trick the object detection and classification algorithms of the AVP system. In order to ensure reliable classification and decision-making in high-risk operational environments, the identified safety requirements seek to strengthen AVP resilience against such adversarial visual complexity.

By examining how spoof objects could impair AVP's performance and make it more difficult for it to differentiate between genuine threats and non-threatening entities, the safety requirements were developed. Second Safety Requirement: Improved Dataset Diversity guarantees that the AVP system is exposed to a range of misleading visual patterns during training, such as different colours, lighting conditions, and reflective surfaces that resemble drones and birds. To categorise and track such intricate visual stimuli, the AVP's machine learning component must be explicitly trained, according to the corresponding ML training requirement.

We establish Safety Requirement 3: Dynamic Perception Algorithms to address the AVP's possible inability to identify unusual spoofing techniques. Adaptive anomaly detection systems that recognise departures from typical visual signatures must be put into place in order to meet this requirement. The AVP's perception model's situational awareness and threat recognition are enhanced by the ML training requirement, which guarantees that it can identify outliers and unidentified spoofing techniques.

To address real-time decision-making vulnerabilities, we also present Safety Requirement 4: Real-Time Adaptive Validation, which requires cross-verification with external air traffic control

(ATC) data in the event that the AVP system encounters uncertain detections. This measure uses real-time data sources to improve validation accuracy rather than requiring extra machine learning training.

We set Safety Requirements 4 and 5 in recognition of possible security threats to guard against adversarial manipulation of AVP datasets. In order to prevent deliberate data corruption that might impair the AVP's capacity to differentiate genuine threats from spoof objects, these requirements impose stringent access control measures and require frequent security audits.

Then, we derive an appropriate set of **ML-component development dataset requirements**:

Dataset requirement structure:

> The [system of interest] ML component [Training/Testing/Black Swan Validation] Dataset shall provide the trainee model with a valuable minimum variety of ...

Table I.31b ML Safety-Training Requirements and Perception Dataset Specifications for AVP

ML Safety-Training Requirements (Training Concept)	**ML Perception development datasets requirements**
ML Safety-Training Requirement 2: The AVP ML component shall be trained to detect, classify, and track a diverse set of complex and deceptive visual patterns (e.g., various colours, lighting angles, reflective surfaces) resembling bird-like objects and drones in controlled airspace. **ML Safety-Training Requirement 3:** The AVP ML component shall be trained to detect deviations from typical visual signatures.	**Datasets Req 2: Bird-Like Shapes:** The AVP perception ML component Training Dataset shall provide the trainee model with a valuable minimum variety of Images and should include bird-shaped objects in various sizes, colours, and poses to represent different flight orientations. **Examples**: Hot-air balloons shaped like large birds (e.g., eagles, hawks, pigeons) with realistic and exaggerated colour schemes. **Datasets Req 3: Drone and Aircraft Replicas:** The AVP perception ML component Training Dataset shall provide the trainee model with a valuable minimum variety of balloon shapes or objects that resemble drones or small aircraft.

	Examples: Objects with quadcopter shapes, fixed-wing outlines, or fuselage designs similar to standard small aircraft. **Pattern deviations**: Objects should feature both matte and reflective surfaces that mimic metallic and plastic finishes to emulate typical drones or aircraft.
	Datasets Req 4: Complicated Visual Patterns and Camouflage The AVP perception ML component Training Dataset shall provide the trainee model with a valuable minimum variety of **Mixed Patterns with Reflective Surfaces**: Images should include objects with highly reflective and mixed-colour patterns that can shift appearance based on angle and lighting. **Examples**: Balloons with alternating dark and light patches, stripes, or polka dots that distort under different lighting angles. **Reflective Materials**: Objects with partially reflective or metallic surfaces that produce glare, mimicking potential interference with sensor recognition.
	Datasets Req 5: complicated colour schemes and contrasts: The AVP perception ML component Training Dataset shall provide the trainee model with a valuable minimum variety of High-contrast images where objects use non-standard, bright colour schemes (e.g., neon, fluorescent) that deviate from typical environmental colours.

	Examples: Brightly coloured balloons with contrasting colours (e.g., bright yellow/black stripes, red/green patches) for the perception algorithms to introduce atypical colour contrasts.
	Datasets Req 6: Environmental Adaptation and Disguise The AVP perception ML component Training Dataset shall provide the trainee model with a valuable minimum variety of **Tree and Cloud Camouflage**: Images should feature objects with visual patterns or textures that could mimic common background elements, such as clouds, trees, or distant landscape elements. **Examples**: Balloons with camouflage patterns in grey or green tones that blend with cloudy skies or forested backgrounds. It also depicts a cloud on the balloon itself. **Object-Background Blending**: Ensure variations of these objects within varied natural settings (forests, urban, and rural areas) to train the system on distinguishing between foreground and background.

I.9 Stage 5: CuneiForm-based Syllabus for Safety-Driven ML Epistemic Intelligence Development[6]

Outcomes from stages 3 and 4 reveal the epistemic uncertainties faced by the architect and their trainee machine regarding the problem domain. The architect must define the real world and

[6] Associated to Section M.7

outline potential unexpected scenarios for the trainee machine to meet objectives. Minimising these uncertainties to a reasonable level (ALARP) is crucial, achieved by creating CuneiForms that capture required training regimes. Thus, this process should be termed ML Epistemic Uncertainty Reduction Training, focusing on enhancing the machine's robustness by increasing the variety of real-world scenarios instead of increasing the variety of image augmentations. Dataset augmentation techniques (e.g., colour modifications) will follow, seen as "Aleatoric Uncertainty Reduction" since they alter pixel randomness rather than actual scenarios.

I.9.1 Step A) Articulate the pictorial problem context:

Datasets Req 6: The AVP perception ML component Training Dataset shall provide the trainee model with a valuable minimum variety of Environmental Adaptation and Disguise

Tree and Cloud Camouflage: Images should feature objects with visual patterns or textures that could mimic common background elements, such as clouds, trees, or distant landscape elements.

Examples: Balloons with camouflage patterns in grey or green tones that blend with cloudy skies or forested backgrounds. It also depicts a cloud on the balloon itself.

Object-Background Blending: Ensure variations of these objects within varied natural settings (forests, urban, and rural areas) to train the system on distinguishing between foreground and background.

Figure I.18 We used DALL-E to generate the this Black Swan Scenario for validation.

Table I.32 CuneiForm Pictorial situation articulation

Pictorial Situation CoT step	**Definition**
Step 1) Define a minimum variety of TOIs and their pictorial appearances	**architect prediction:** the architect asserts that the perception system may face a pictorial situation with: A hot-air balloon with sky and clouds painted on its fabric.
Step 2) Consider a minimum variety of other objects that TOIs aim to influence	**architect prediction:** the architect asserts that: the hot-air balloon aims to influence the discoverability of itself by AVP.
Step 3) Consider a minimum variety of objects that TOIs must appreciate	**architect prediction:** the architect asserts that: the hot-air balloon may have to appreciate the following environmental **scenery aspects:** **Cloud Cover Variability**: The balloon should account for various cloud formations and densities, as clouds could enhance or diminish its camouflage. **Sky Colour and Light Conditions**: Different times of day or weather conditions will affect how the sky appears. The balloon's camouflage pattern may need to appreciate transitions from sunrise to sunset, clear skies, and changing hues.
Step 4) Consider a minimum variety of what other objects TOIs must control the correctness of predicting their shapes.	**Architect prediction:** The architect asserts that the hot-air balloon must carefully control its visual presentation to maintain inconsistent shape predictability, reducing the likelihood of correct classification or shape clarity within the AVP system's detection model. This includes: Varying the rotation of the balloon in order to increase the randomness of its reflective

	glossy surfaces, thus causing a maximum likelihood of AVP perception failure.
Step 5) Produce pictorial problem context.	**Pictorial problem context:** The architect asserts that the AVP perception system may face a pictorial situation with: A hot-air balloon{1} with a sky and clouds painted on its fabric{2}. A background of varying cloud formations and densities{3}. The balloon must adjust for changes in sky appearance based on different times of day and, including transitions from sunrise {4} to sunset{5}, clear skies{6}, and variable light hues{7}. The balloon varies its rotation {8} to increase randomness in its reflective glossy surfaces {9}, maximising the likelihood of AVP perception failure.

I.9.2 Step B) Characterise the Training Classes for CuneiForms:

In this step, we will define one CuneiForm that satisfies the pictorial situation context. We will consider a maximum of 5 objects in this Cuneiform. In this case study, we will need to define the range of general aircraft 3D orientation specification. We choose to define 18 possible variations:

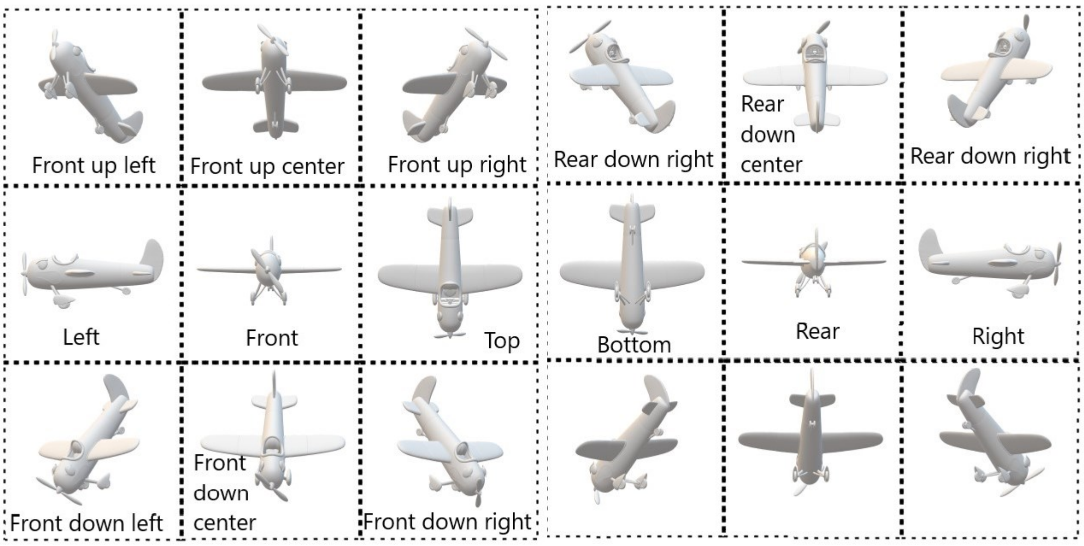


Figure I.19 general aircraft 3D orientation specification

In this step, we will define one CuneiForm that satisfies the context of the pictorial situation. We will consider a maximum of 5 objects in this Cuneiform.

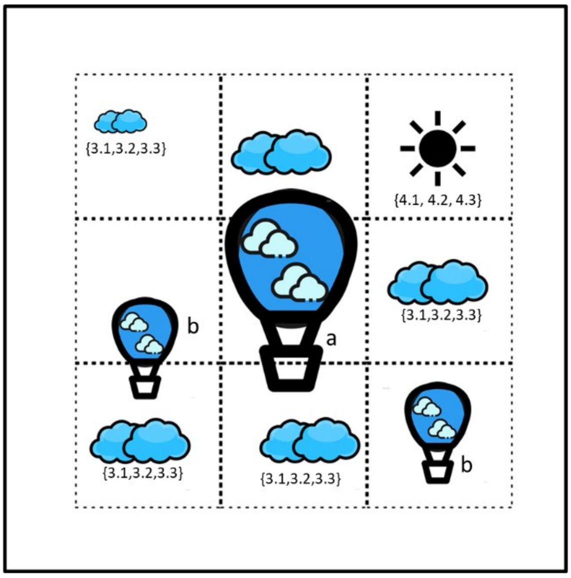
Table I.33 Characteristic Training Classes definitions for a CuneiForm abstract image

CuneiForm characteristic	**Definition**
Step 1) Define visible horizon attitude.	No Horizon
Step 2) Define all the TOIs and their aesthetic complexity. Then, generate abstract representative icons for the CuneiForm abstract image.	Several hot-air balloons {1.1}, sky and clouds painted on its fabric {2.1}
Step 3) Define TOI's motion trajectory and dynamic optical situations. Then, update the generated abstract representative icons for the CuneiForm abstract image.	**Motion trajectory:** static, no motion {1.2}. **Dynamic optical situation:** captured without optical blur {1.3}.
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.	Clouds, Mostly Clear or Sunny (1/8 to 2/8 oktas) {3.1}, sphere of the sun during day {4.1}
Step 5) Define the background Objects' Motion situations and dynamic optical situations. Then, update the generated abstract representative icons for the CuneiForm abstract image.	**Background objects' motion trajectory:** static {3.2, 4.2} **Dynamic optical situation:** no motion blur {3.3, 4.3}
Step 6) Define TOI's positioning in the CuneiForm. Then, generate	Any {1.4}

abstract representative icons for the CuneiForm abstract image.	
Step 7) Define TOI's 3D orientation. Then, update the generated abstract representative icons for the CuneiForm abstract image.	Front {1.5}
Step 8) Define the optical distance for each TOI in nindans. Then, update the generated abstract representative icons for the CuneiForm abstract image.	A balloon is represented in a pictorial distance of 3 nindan (equivalent to 3/9 of the total area of the pictorial frame) {1.6} A balloon is represented in a pictorial distance of 1 nindan (equivalent to 1/9 of the total area of the pictorial frame) {1.7}
Step 9) Design the relevant icons to produce the CuneiForm and give an example of an instantiating image	Hot-air Balloon:  , Clouds: , Sun: 

I.9.3 Final CuneiForm

An output cuneiform would be:



a: {1,1.1,1.2,1.3,1.4,1.5,1.6, 2.1, } b: {1,1.1,1.2,1.3,1.4,1.5,1.7, 2.1, }

Figure I.20 Example CuneiForm and instantiated image

An example of CuneiForm non-compliant images may be:

Table I.34 Examples of non-compliant instantiations

CuneiForm non-compliant Image	**Reasons**	**Decision**
	Such an image would not pass the CuneiForm-I.20 characterisation, as it inadequately captured the presence of clouds and the sun's position on the opposite side. Nonetheless, despite this shortcoming, the image illustrates a rare event. Consequently, it may be considered for inclusion in validating a new variation of CuneiForm.	To be added to the Black Swan validation datasets.
	Such an image would not pass the CuneiForm-I.20 characterisation, as it inadequately captured the presence of cloud paintings on the balloon. Nonetheless, despite this shortcoming, the picture illustrates a rare event. Consequently, it may be considered for inclusion in validating a new variation of CuneiForm.	To be added to the Black Swan training dataset, to instantiate another Black Swan CuneiForm training class.
	Such an image would not pass the CuneiForm-I.20 characterisation, as it inadequately captured the presence of cloud paintings on the balloon. Nonetheless, despite this shortcoming, the image illustrates a rare event. Consequently, it may be considered for inclusion in validating a new variation of CuneiForm.	To be added to the Black Swan training dataset, to instantiate another Black Swan CuneiForm training class.

 [Link](#)	Such an image would not succeed in passing the CuneiForm-I.20 characterization, as it does not capture the presence of cloud paintings on the balloon. However, it certainly captures a rare or seasonal event.	To be added to the Black Swan validation datasets, to instantiate another Black Swan CuneiForm validation class.
 [Link](#)	Such an image would not succeed in passing the CuneiForm-I.20 characterisation, as it does not capture the presence of cloud paintings on the balloon. Also, the situation in the image is clear of any cloud.	To be discarded as it does not necessarily add valuable information in comparison to the above.

I.9.4 Develop the Training, Testing and Black Swan Validation Datasets

The final step of this process would be gathering various images that conform to the CuneiForm. In this exercise, we will assume that the architect has done the due diligence per their data generation methods. Instead of gathering and generating a dataset, we will validate the AVOIDDS training strategy against the CuneiForm framework and training classes coverage criteria.

I.10 Stage 6: Black Swan-driven ML Development and Testing[7]

Usually, there would be the model training process begins. We will assume that a model was developed, and the client needs to validate the coverage of the training dataset. So, in this stage, we will use the CuneiForm method to validate the quality of a pre-existing dataset named AVOIDDS. This repository contains datasets, models, and simulators for the AVOIDDS (Aircraft Vision-based Intruder Detection Dataset and Simulator) benchmark, which centres around the vision-based aircraft detect-and-avoid (DAA) problem. The full AVOID dataset of 72,000 samples is available here: purl.stanford.edu/hj293cv5980.

Our validation methodology was based on covering the CuneiForm framework for Epistemic Training Strategy, which allows for structured epistemic analysis of dataset completeness by categorising data based on key dimensions, including:

- Time-of-day distribution.
- Cloud type.
- Horizon orientation.
- TOI's pictorial positioning zones.
- TOI's pictorial distance.

Key Assumptions:

We will use the following definitions:

Concept	Definition
ALARP	As Low As Reasonably Practicable
ArcUC	Architect Epistemic Uncertainty Curve for a given dataset
PHI	Pictorial Visible Horizon Attitude Indicator
TOI	Target Object of Interest

- The sample_small folder of training and validation datasets is statistically representative of AVOID dataset (73,000 instants).

[7] Associated with Section M.8

    - 30 images, training sample.
    - 30 images validation sample.

- Time of day, Clouds types, file names all taken directly from the provided situation_data.xlsx in the repository.
- The CuneiForm coverage results on the representative sample are accepted as a quality assessment of the entire AVOID dataset.
- For this exercise, we needed to define TOI's 3D orientation categories for completeness to achieve coverage or 3D orientation and minimise epistemic uncertainty ALARP. To such end, we define the following categorical system of airplane orientation training classes:

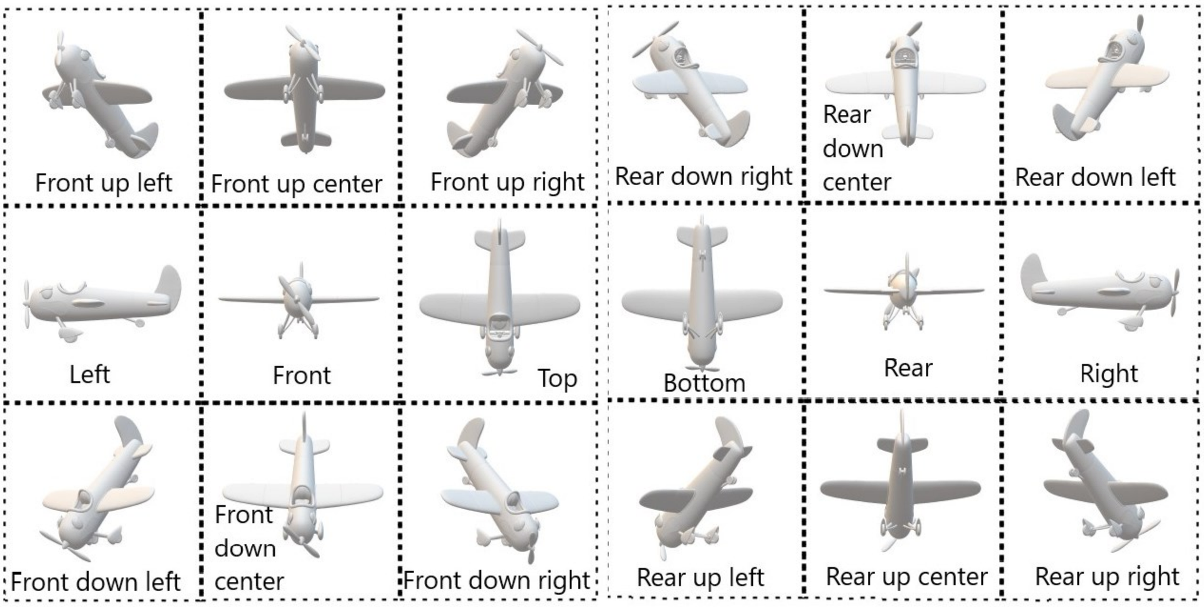


Figure I.21 aeroplanes 3D orientations categories (training classes)

Figure I.21 presents a structured classification system for defining aircraft's three-dimensional (3D) orientation categories within a training dataset. This classification aims to cover various orientations comprehensively, ensuring minimisation of epistemic uncertainty to As Low as Reasonably Practicable (ALARP) levels. This structured approach improves dataset completeness in training AI-based perception models in airborne object detection and classification tasks.

The figure systematically categorises airplane orientations by organising them into distinct viewpoints based on azimuthal and elevational perspectives. These orientations are essential for ensuring that trained AI models generalise effectively across various real-world aircraft configurations. There are four main reference perspectives within the classification system:

1. Frontal Perspective (Front, Front-Up, Front-Down)
    - records changes in how the aircraft is viewed from the front.
    - comprises both standard frontal orientation and views that are either elevated or depressed at various angles (e.g., Front Up Left, Front Up Right, Front Down Centre).
2. Rear Perspective (Rear, Rear-Up, Rear-Down)
    - depicts an aircraft from the back, including standard rear-facing angles and different elevations.
    - crucial for identifying aircraft that are retreating or in trailing situations that present difficulties for detection algorithms.
3. Lateral Perspective (Left, Right)
    - Includes aircraft viewed from the side, either left or right.
    - Ensures the model can recognise aircraft from side angles, which is critical in multi-aircraft detect-and-avoid scenarios.
4. Top-Down and Bottom-Up Perspectives
    - captures aeroplanes directly from the top or bottom, for example.
    - These viewpoints are especially difficult for vision-based models because they frequently look like silhouettes against the sky or ground projections.

The method guarantees that the AI system is exposed to a comprehensive representation of potential aircraft views in 3D space by organising the training dataset around these 16 orientation categories. In vision-based models for airborne object detection, tracking, and avoidance in dynamic operational environments, this methodology is crucial for improving generalisation and lowering epistemic uncertainty.

Additional robustness against biases that may result from incomplete viewpoint exposure in training datasets is provided by the incorporation of elevational shifts (up, down) and lateral tilts (left, right). If these different orientations are not taken into consideration, the model may perform inconsistently in real-world situations where dynamic flight manoeuvres cause aircraft positions to vary greatly.

I.10.1 CuneiForm Training Strategy as a Validation Process for Datasets

We will use some of the training classes we defined in section **E.7: CuneiForm ML Epistemic Training Strategy** to reference what needs to be covered in the AVOID training dataset. The following is a snapshot of how the training sample looks:

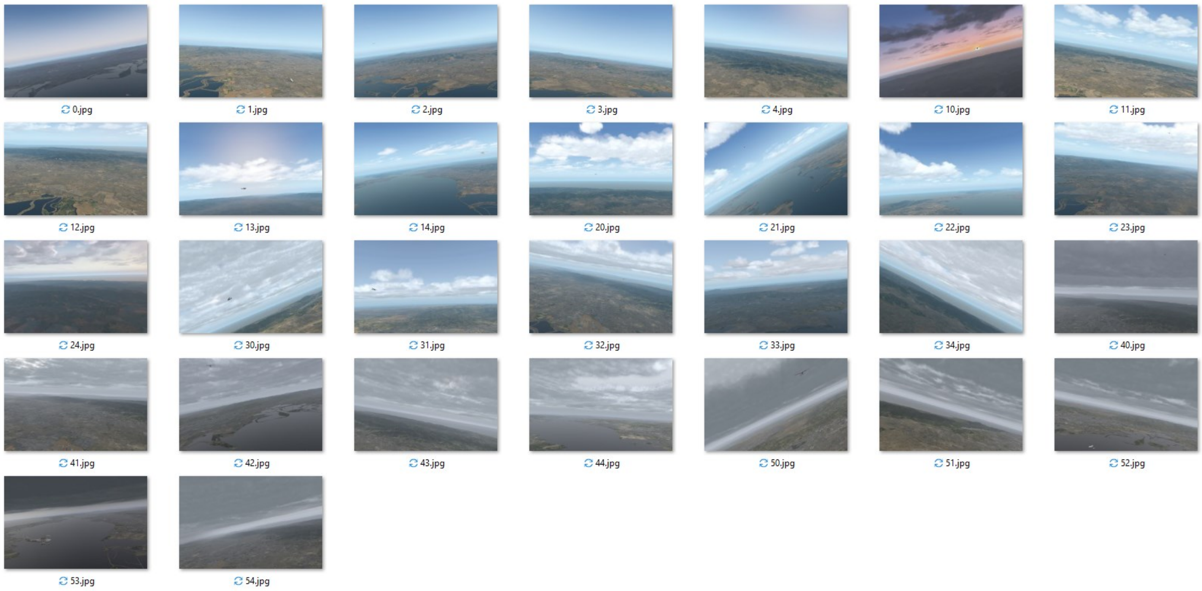


Figure I.22 AVOID training dataset random sample

Figure I.22 shows the set of images we used to evaluate the quality of the AVOID training dataset in terms of coverage using CuneiForm epistemic dimensions coverage. In other words, we look at the dataset and examine any imbalance of coverage against the categories for some of the CuneiForm dimensions. In this exercise, we included three CuneiForm Training classes in our examination and produced an examination report, which can be found in here:

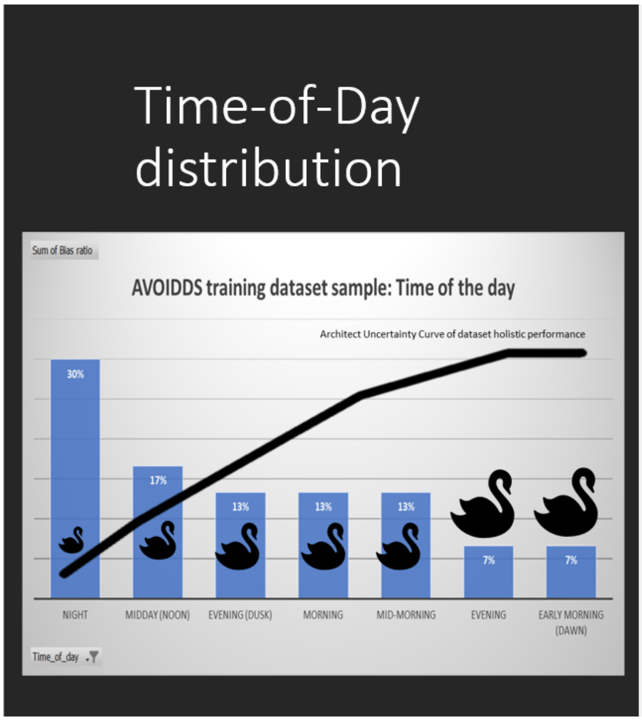
- Pictorial Horizon orientation.
- TOI's pictorial positioning zones.
- TOI's pictorial distance.

We also included one aspect of our ODD, the Time-of-Day categorisation. AVOIDDS provides a balanced (uniformly distributed) distribution of information about the types of clouds.

The validation process was structured into the following key steps:

I.10.1.1 Examining AVOIDDS Training Strategy in Covering Time-of-Day Training Classes

Our examination discovered that there exists an imbalance in coverage, with 47% of instances recorded at night and midday (noon) demonstrating strong bias. The lack of balanced exposure constitutes a potential for exponentially unpredictable, high-risk emergent Black Swan behaviour performed by the perception. The dataset does not satisfactorily meet the ALARP requirement. Therefore, we have sufficient reason to believe that this dataset does not pass the ALARP criteria for covering time-of-day categories. The unbalanced coverage leads to hazardous uncertainty in the intelligent system's behaviour during Black Swan scenarios in a respective category. The figure below captures the results of examining AVOIDDS time-of-day coverage based on their sample data.



- The following is the training dataset strategy of AVOIDDS sample dataset in covering time of day dimension:

Category	Time Range
Early Morning (Dawn)	4:00 AM - 6:00 AM
Morning	6:00 AM - 9:00 AM
Mid-Morning	9:00 AM - 11:00 AM
Midday (Noon)	11:00 AM - 1:00 PM
Afternoon	1:00 PM - 4:00 PM
Evening	4:00 PM - 6:00 PM
Evening (Dusk)	6:00 PM - 8:00 PM
Night	8:00 PM - 4:00 AM

Time of day class	instants count	Class probability
Night	9	30.00%
Midday (Noon)	5	16.67%
Evening (Dusk)	4	13.33%
Morning	4	13.33%
Mid-Morning	4	13.33%
Early Morning (Dawn)	2	6.67%
Evening	2	6.67%
	30	

Figure I.23 AVOIDDS Time-of-Day coverage validation report based on the training sample folder

Figure I.23 presents our examination of the time-of-day coverage within the AVOID training dataset, highlighting critical imbalances in its distribution. The dataset exhibits a strong bias towards night and midday (noon) instances, which account for 47% of the recorded samples. This skewed representation results in an incomplete epistemic coverage, introducing potential hazardous uncertainty when the model encounters underrepresented time-of-day conditions in real-world applications.

The figure employs a Black Swan icon size to visually depict the degree of epistemic uncertainty across different time-of-day categories. Larger swan icons indicate higher uncertainty, suggesting that the dataset is less reliable in those categories and that the trainee model is likely to exhibit erratic or unpredictable behaviour in those underrepresented conditions. Specifically, early morning (dawn), evening and mid-morning classes exhibit a notably low representation, increasing the likelihood of exponential performance degradation in these unseen scenarios.

From a safety-critical AI validation perspective, the dataset's coverage fails to meet the ALARP (As Low As Reasonably Practicable) criterion, as the uneven distribution introduces a significant risk of Black Swan emergent behaviours. The absence of balanced exposure across all time-of-day categories weakens the model's ability to generalise and respond effectively in diverse lighting and environmental conditions, which is crucial for robust aircraft detection-and-avoid applications.

Therefore, the findings illustrated in Figure I.22 underscore the importance of a more balanced dataset composition. Without further rebalancing or augmentation of underrepresented time-of-day classes, the AVOID dataset remains susceptible to uncertainty, reducing its reliability and safety for real-world deployment in dynamic aviation environments.

I.10.1.2 Examining AVOIDDS Training Strategy in Covering Clouds type

The dataset was evaluated for **balanced exposure to different cloud formations**. While it passed the ALARP (As Low As Reasonably Practicable) requirement for general cloud-type coverage (Figure I.24), if we consider the Time-of-Day per cloud-type distribution, we discover potential biases in the dataset (Figure I.24). The figure below reflects the training sample cloud-type alone coverage:

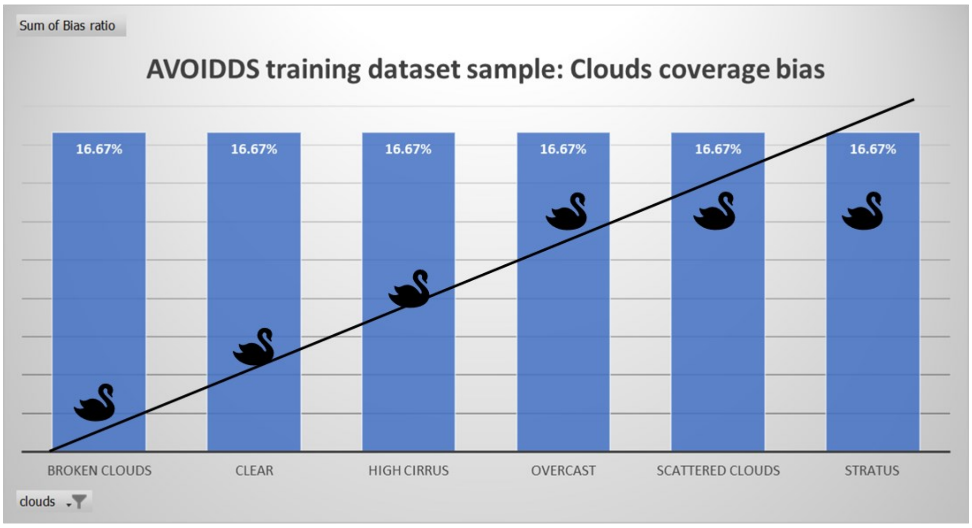


Figure I.24 types of cloud coverages examination

The AVOID training dataset has a balanced coverage of all types of clouds. Figure I.24 presents a high-level overview of the dataset's cloud-type distribution, indicating that each cloud category—broken clouds, clear sky, high cirrus, overcast, scattered clouds, and stratus—is covered equally at 16.67%. This uniformity suggests that, at first glance, the dataset satisfies ALARP (As Low As Reasonably Practicable) requirements regarding cloud-type representation.

We then examined a more refined approach that examines the type of clouds and type of day. Below is the histogram that showcases the types of missing scenarios in the training dataset sample:

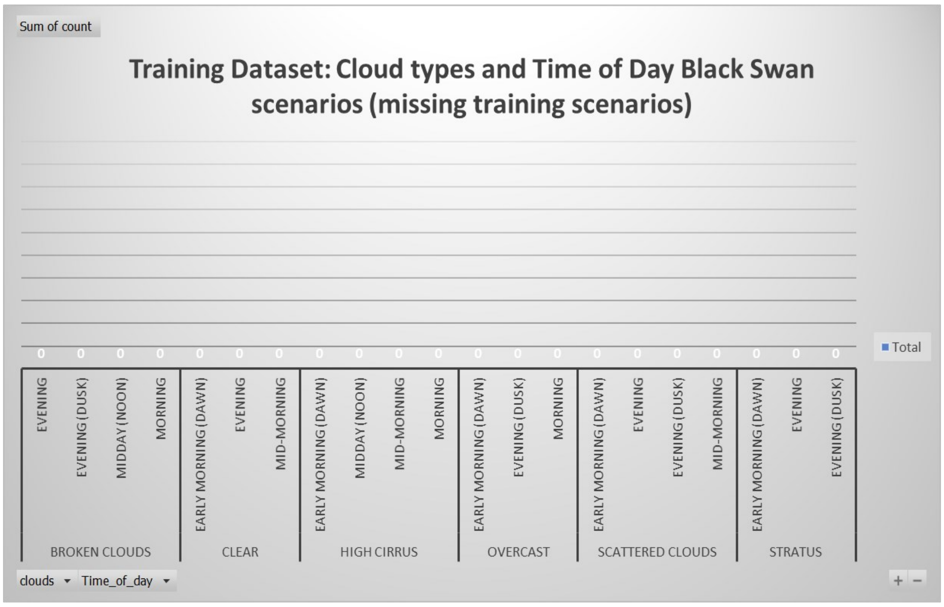


Figure I.25 Potential epistemic confusion in Black Swan Scenarios

Figure I.25 captures a more refined description of potential real-world scenarios in which we have sufficient reason to lose trust that any ML component trained on the AVOIDDS would detect objects' reliability if faced with the Black Swan Scenario in those conditions. In other words, Figure I.25 extends this analysis by examining the intersection of cloud types with time-of-day variations, highlighting the absence of several crucial Black Swan scenarios. The missing instances indicate conditions in which the dataset does not train the model adequately, increasing epistemic uncertainty. If an ML perception system is trained on AVOIDDS but deployed in a scenario where, for instance, a broken-cloud sky occurs at dusk—an untrained combination—there is an increased risk of unpredictable model behaviour, potentially leading to failures in object detection.

In contrast, the training dataset sample reveals the following combined distribution between cloud and Time-of-Day in Figure I.25. It shows that AVOIDDS has a training strategy that covers scenarios such as:

- Broken cloud sky during dawn.
- Clear sky during morning.
- High cirrus clouds during the night.
- Overcast clouds during the evening.
- Scattered clouds during midday.
- Stratus cloud during morning...etc.

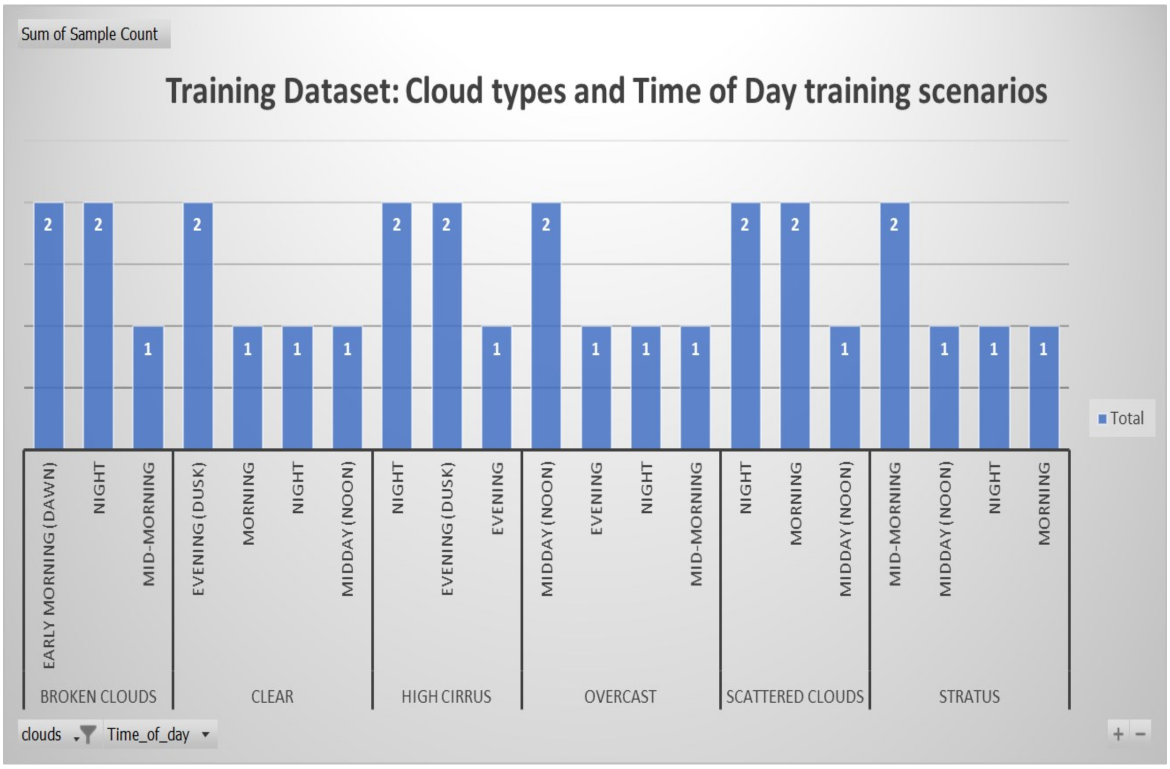


Figure I.26 Example scenarios covered by AVOIDDS training strategy

Figure I.26 further refines this evaluation by showcasing the specific cloud-time-of-day pairings in the dataset. It demonstrates that while the dataset covers some expected scenarios, such as high cirrus clouds at night and scattered clouds at midday, it remains incomplete for ensuring robust real-world performance. The analysis confirms that while the dataset's cloud coverage is theoretically balanced, its intersection with time-of-day conditions introduces significant epistemic uncertainty, limiting the reliability of any trained ML model under novel operational conditions.

I.10.1.3 Examining AVOIDDS Training Strategy in Covering Pictorial Distance Training Classes

See sections 4.5.5.5 and 5.9.5 for a background definition of this training class. The dataset was tested for coverage across different object distances (close, medium, far, and extremely far). A major imbalance was found, with 50% of instances covering only 20% of the required categories. Additionally, no instances were classified as "dangerously close distances," which could cause critical failures in AI-based collision avoidance models. See Figure below:

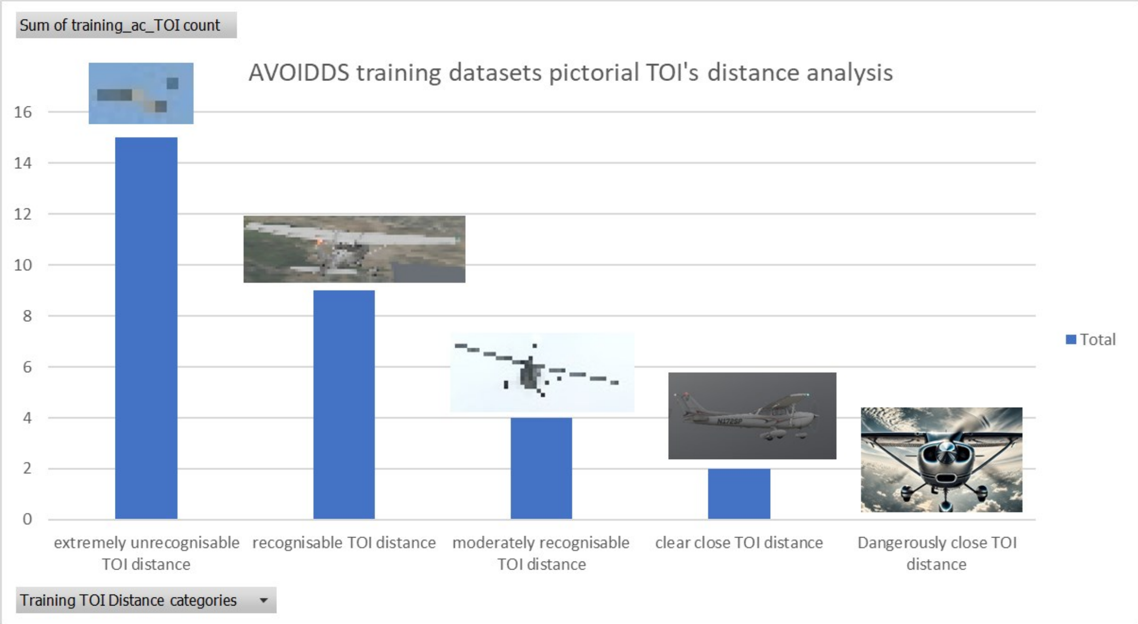


Figure I.27 TOI distance categories distribution

Figure I.27 presents our examination of the AVOID training dataset's coverage of TOI distance training classes, revealing a significant imbalance in the representation of different distance classes.

Regarding trustworthiness in facing Black Swan scenarios, the distribution indicates more uncertainty (less trust) about the "dangerously close distance" category. Signified by the size of the Black Swan icon in the figure below:

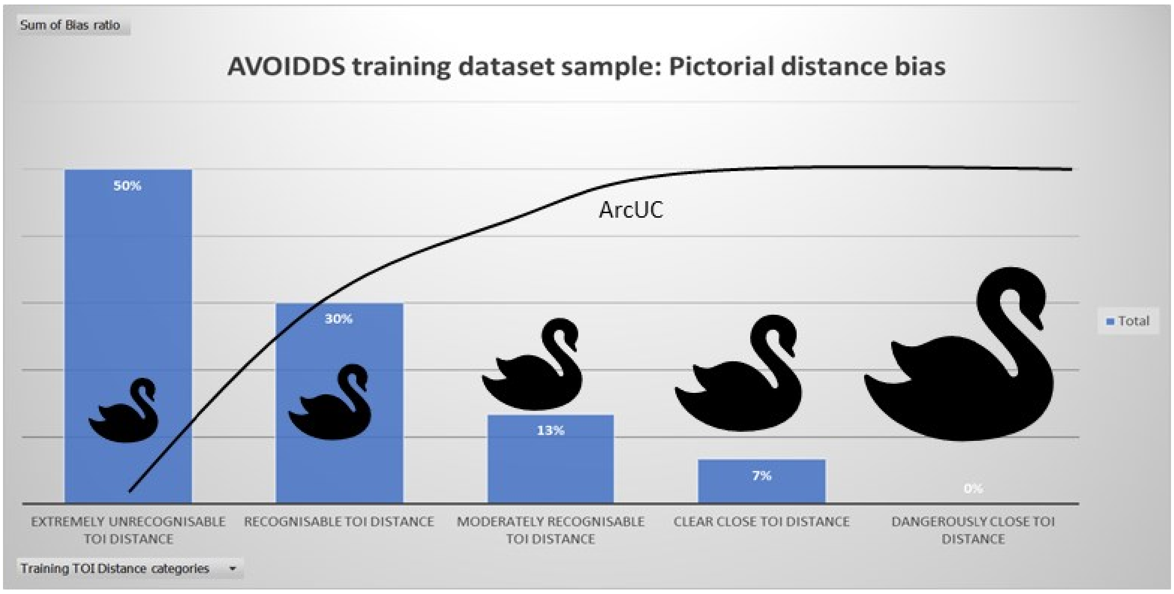


Figure I.28 Trust in performance in facing Black Swan Scenarios related to a respective category.

Figure I.287 shows how the architect's uncertainty increases as the likelihood of unreliable performance during a Black Swan event in a given category increases. It is interesting to notice that AVOIDDS did not consider a dangerously close aeroplane for a safety-critical application such as avoiding mid-air collisions. Considering that some incidents happened because two aeroplanes came dangerously close to each other without the pilots noticing, see the following [2]. Consequently, the dataset does not satisfactorily fulfil the ALARP requirement for pictorial distance coverage.

I.10.1.4 Examining AVOIDDS Training Strategy in Covering TOI's Positioning Training Classes

See section E.7.4.4 for a background definition of this training class. The dataset's representation of **object positioning in the visual frame** was analysed. We found that the dataset exhibits a significant imbalance, with 60% of perception experiencing TOIs in only 20% of the possible positioning quadrants. Notably, there is a complete absence of experience in the following possible positions:

1. up center/center
2. up left/center left
3. center left/center
4. up left/up center
5. down left/down centre

The lack of exposure constitutes a potential for exponentially unpredictable, high-risk emergent Black Swan behaviour performed by the perception. Figure I.28 demonstrates the lack of coverage in the listed positions, indicating potential sources of sufficient concerns in facing black Swan Scenarios in such categories. The histogram shows a heavy concentration of training instances in central and slightly off-centre positions, with the highest number of attentive mentions recorded at "centre" (17%) and "downright" (13%). Other positioning categories, such as "center left," "center right," "down left," and "down right," receive moderate representation (7%), while numerous other positional categories have significantly lower representation (3%). The Black Swan icons in the figure depict the increasing uncertainty in model reliability as the training dataset lacks exposure to key TOI positional scenarios. Most notably, the dataset completely lacks experience in five critical positions: "up center/center," "up left/center left," "center left/center," "up left/up center," and "down left/down center". The absence of these positioning categories presents a risk in real-world object detection, particularly in aviation safety applications, where TOIs may appear in any arbitrary position relative to the observer. The dataset's failure to comprehensively train AI models across the full spectrum of TOI positions indicates a significant epistemic gap, increasing the likelihood of unpredictable Black Swan failures in operational environments.

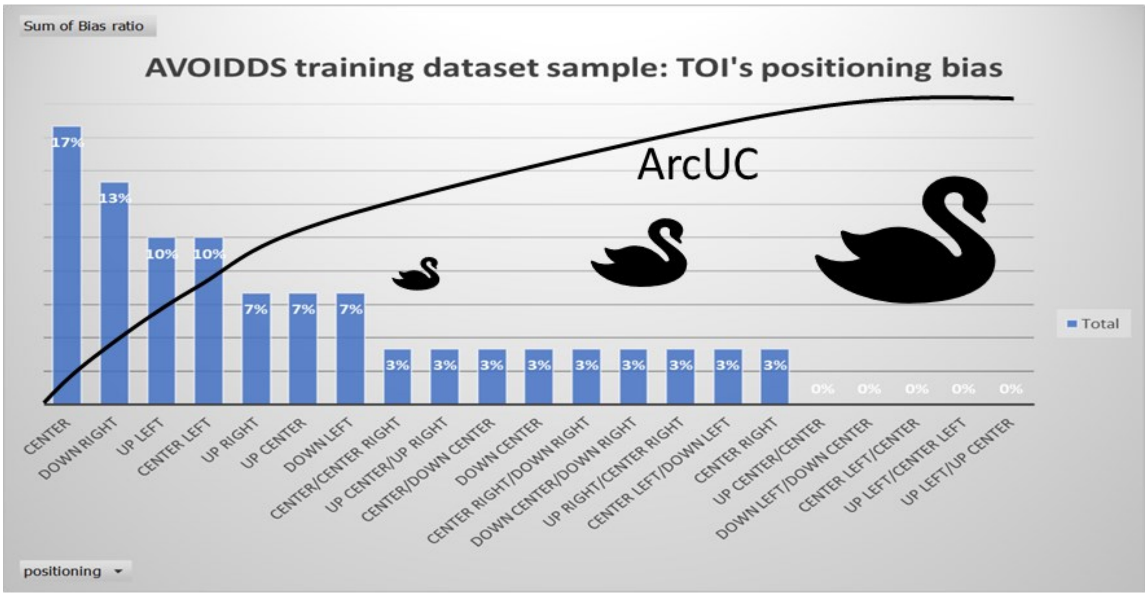


Figure I.29 TOI's positioning training classes coverage.

Figure I.30 shows which training scenarios the AVOIDDS epistemic training strategy missed in its training dataset to illustrate the missing training classes clearly. Red zones are the areas in images that the training dataset is missing, which gives us sufficient reason to suspect that the trained ML may struggle with scenarios where TOIs appear to be in those regions.

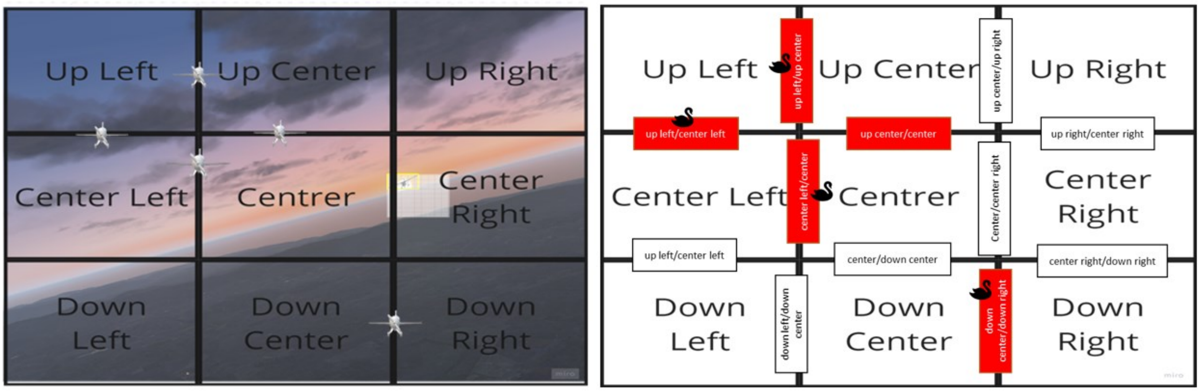


Figure I.30 visualising the areas of missing scenarios in AVOIDDS training strategy

Consequently, the dataset does not satisfactorily fulfil the ALARP (As Low As Reasonably Practicable) requirement for TOI’s positioning coverage.

I.10.1.5 Examining AVOIDDS Training Strategy in Covering Pictorial Horizon Attitude Training Classes

See sections E.7.5.3, 4, 5 for a background definition of this training class. Our examination of coverage against Visible Horizon Attitudes training classes showed that AVOIDDS training strategy exhibits significant imbalance, with 60% of perception experiencing TOIs in only 23% of the possible horizon attitudes. Notably, there is a complete absence of experience in the following possible horizon attitudes:

- Level Horizon
- Negatively Tilted Elevated Horizon
- Acute Angled Bird's Eye Ground View
- Bird's Eye Ground View
- Ascending Rocket Sky View
- Negatively Tilted level Horizon
- Acute Angled Rocket Sky View

The lack of exposure constitutes a potential for an exponentially unpredictable high-risk emergent Black Swan behaviour performed by the perception.

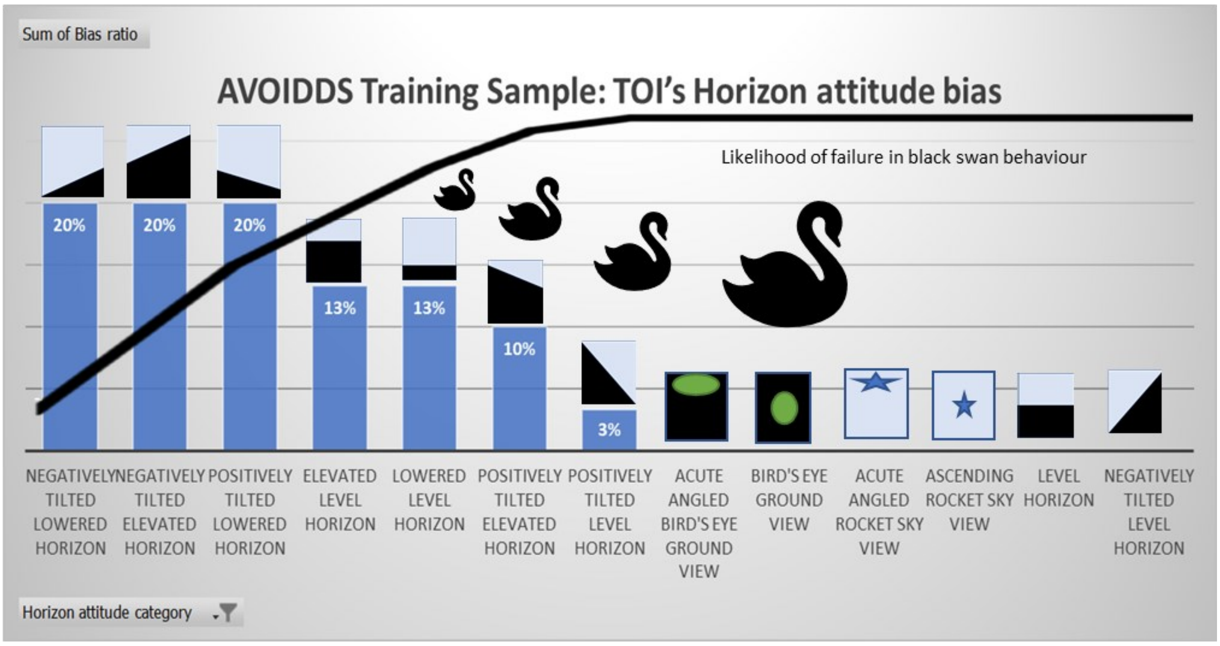


Figure I.31 AVOIDDS training strategy coverage of Horizon Attitudes training classes. The icons on top of the pins depict the nature of the horizon attitude and how the TOI would look.

Figure I.31 presents the analysis of the AVOID training dataset's coverage of TOI horizon attitudes, revealing substantial biases in exposure to different orientation perspectives. This distribution results in a considerable epistemic gap, decreasing our trust in the dataset to enable the trainee model handling Black Swan scenarios where TOIs appear in underrepresented attitudes. The Black Swan markers indicate the rising architect uncertainty about AVOIDDS quality, allowing any trained model to generalise to unseen cases. Consequently, the dataset needs to satisfactorily fulfil the ALARP (As Low As Reasonably Practicable) requirement in terms of horizon attitude coverage.

I.10.2 Possible Missing Pictorial Hazards Identified in AVOIDDS

Our examination of the AVOID dataset has revealed limitations that affect its suitability for real-world safety-critical applications, particularly in mid-air collision avoidance perception systems. While AVOIDDS is a useful benchmarking tool, it lacks the robustness to ensure reliability under complex operational conditions. The identified hazards highlight areas where the dataset introduces epistemic uncertainty, increasing the likelihood of perception failures. Table I.35 outlines the key hazards and associated mitigations necessary to enhance the safety and reliability of AVOIDDS to enable a reliable ML model.

One of the primary failure modes identified is ambiguity in object identification, where the perception system may struggle to correctly recognise aircraft that are either at long distances or appear from unconventional orientations. This ambiguity poses a critical risk, as misinterpretation of another aircraft's position and movement can lead to incorrect avoidance actions. To mitigate this risk, the perception system must be trained on various aircraft orientations, including atypical perspectives such as inverted or tilted positions, which may arise due to dynamic flight manoeuvres or environmental factors.

From an assurance perspective for real-world application, to ensure that the datasets accurately reflect the operational domain, the architect must provide objective evidence that can be quality assured by third parties. The AVOIDDS sample indicates the presence of images that human validators cannot easily verify to determine whether those images cover a particular risk. For example, image 24.jpg part of the training sample, includes a TOI where it is unclear which 3D orientation it is captured in. See the Figure below:

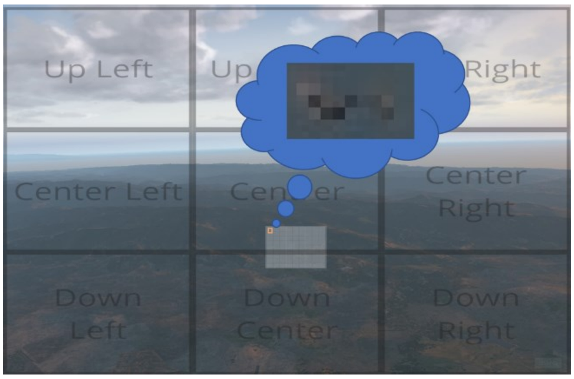


Figure I.32 a sample training image 24.jpg in AVOIDDS dataset where the TOI's 3D orientation is unrecognisable

The figure above showcases a weakness in the dataset, in that there are TOIs that are too far, so much so they may have a distorted visual appearance. This unrecognisable visual appearance may lead the model to an increased number of recall (false positives).

Another critical issue (can be claimed to have been attempted to be addressed by the AVOIDDS) is difficulty recognising distant or low-visibility objects, particularly in adverse conditions such as night-time, poor lighting, or heavy cloud cover. Aircraft appearing in such situations may not be effectively detected by a model trained on AVOIDDS, which lacks sufficient representation of these challenging scenarios. The system should be improved to recognise objects at great distances and in a variety of atmospheric conditions in order to overcome this limitation. This involves managing objects with low contrast and those that are partially hidden by atmospheric scattering or clouds, among other environmental factors, to ensure accurate detection even under less-than-ideal circumstances.

Uncertainty in aircraft orientation and behaviour presents another drawback, as the system might not accurately interpret an aircraft's attitude and trajectory. In high-risk situations, like two planes crossing paths or one making an unexpected manoeuvre, this problem is especially troubling. The perception system may act inappropriately or slowly, increasing the chance of a collision, if it is unable to discriminate between a steady flight path and an unpredictable movement. This can be lessened by training the model on a wide variety of aircraft orientations and dynamic flight behaviours, which will guarantee that it can predict trajectories with accuracy and differentiate between typical and unusual flight patterns.

Table I.35 Hazards and mitigations associated with AVOIDDS

Missing Perception Failure mode	**ML Safety-Training Requirement (Training concept):**
Ambiguity in Object Identification: **Risk**: The system might fail to recognise other aircraft when they are at a very close distances or have an orientation that makes them difficult to identify. For example, if an aircraft is coming from an unconventional angle (e.g., upside down or tilted), the system might misinterpret its position and distance, leading to incorrect avoidance actions.	**Mitigation**: The perception system shall be trained to handle diverse orientations, particularly those that might appear unconventional due to the aircraft's flight dynamics or environmental conditions.

Difficulties in Recognizing Distant or Low-Visibility Objects: **Risk**: In low-light (e.g., night), weather, or long-range scenarios, the system might fail to detect aircraft that are at extreme distances or partially obscured by clouds, smoke, or other atmospheric factors. The system's performance could degrade, especially in critical situations where detection and timely reaction are necessary to avoid collisions.	**Mitigation**: The system should be trained to handle ambiguous or partial visual information by recognising objects at varying distances and through weather conditions like broken clouds or low visibility. Proper handling of low-contrast or poorly illuminated objects will ensure that the system remains effective even in suboptimal conditions.
Uncertainty in Aircraft Orientation and Behavior: **Risk**: Due to their flight path (e.g., ascending, descending, tilted), aircraft can appear in various orientations. If the perception system cannot accurately determine the orientation or trajectory of another aircraft, it might misinterpret its movement or predict the wrong course of action. This is particularly dangerous when aircraft are near each other or crossing paths.	**Mitigation:** The system must be trained to interpret diverse aircraft orientations, especially in scenarios where they might not be level, and to account for dynamic flight behaviour. This would ensure that it can differentiate between aircraft in a stable flight path versus those in unusual orientations, thus avoiding collision.

I.10.3 Retrospective Production of CuneiForm Abstract Images

In our examination of the AVOID dataset, we retrospectively reconstructed potential CuneiForms that may have been implicitly guiding the dataset's training strategy. This process involved analysing the dataset's sample characteristics regarding object positioning, motion, environmental context, and horizon attitude and mapping them onto an abstract CuneiForm framework. The objective was to determine what the dataset's composition would be like if we were to generate a CuneiForm abstraction.

To reconstruct the CuneiForm representations, we extracted their key dimensions by examining a selection of instantiated training images, such as cessna_ac_training50 through cessna_ac_training54. The reconstruction involved segmenting each image according to the CuneiForm dimensional framework, which captures essential training characteristics. These dimensions included TOI (Target of Interest) pictorial positioning, TOI's 3D orientation, TOI pictorial distance, and visible horizon attitude. The following is the example CuneiForm Training Strategy framework that covers training images 50 to 54:

Table I.36 Example CuneiForm Training Strategy

Abstract CuneiForm Characteristics (dimensions)	**Abstract CuneiForm Characteristics definitions**
TOIs definition and their aesthetic complexity	Single-engine propeller aeroplane {1}
TOI Motion and Dynamic optical situations	**Motion trajectory:** Linear motion captured in consecutive images where the aeroplane appears to move in a straight line at a constant speed (no acceleration) {1.2}. **Dynamic optical situation:** captured without optical blur {1.3}.
Background Objects associated with TOIs	Stratus Clouds{2} green-terrain {3} water surface{4}
Background Objects Motion and Dynamic optical situations	Background objects' motion trajectory is static {2.1,3.1,4.1} Dynamic optical situation: no motion blur{2.2,3.2,4.2}
Visible horizon attitude	Negatively Tilted Lowered Horizon{5} Positively Tilted Lowered Horizon{6} Negatively Tilted Elevated Horizon{7}
TOI's Pictorial Positioning	center left/down left{1.4} center/down center{1.5} down left{1.6} up center/up right{1.7}
TOI's Pictorial Distance	recognisable TOI distance{1.9}, clear close TOI distance{1.10}, extremely unrecognisable TOI distance{1.11},
TOI's 3D Orientation	Front{1.12} rear down right{1.13} Right{1.14}

	Unknown{1.15}

Table I.37 presents an Example CuneiForm Training Strategy, which defines a structured framework for training an AI perception model using a well-defined dataset. This framework ensures comprehensive epistemic coverage by categorising essential visual and spatial characteristics of the TOI and its surrounding environment. Each row in the table corresponds to a specific CuneiForm dimension, encapsulating key elements that define an object's training exposure within the dataset.

The dataset's main subject of interest, a single-engine propeller aeroplane, is established by the TOI's definition and aesthetic complexity. This guarantees that the AI model's object classification is followed by every training instance. The anticipated movement behaviour of the aircraft is described in the TOI Motion and Dynamic Optical Situations section. Each captured frame maintains a structured and predictable optical flow thanks to the TOI's linear motion trajectory, which does not accelerate. Furthermore, the dynamic optical condition is meticulously managed to avoid synthetic motion blur, guaranteeing feature extraction clarity.

The dataset's environmental components that go with the aircraft are described in detail in the Background Objects Associated with TOIs dimension. This offers a range of natural settings, such as water surfaces, green terrain, and stratus clouds. The TOI stays the main focus of training data since the motion of background objects and dynamic optical conditions correspondingly verify that these elements stay static with no discernible motion blur.

Tilted horizon categories, such as negatively tilted lowered, positively tilted lowered, and negatively tilted elevated horizons, are introduced by the Visible Horizon Attitude dimension. These variations replicate various aerial perspectives that an AI-based vision system might come across, reflecting actual flight conditions.

The location of the aeroplane in the frame is specified by the TOI's Pictorial Positioning dimension. Positions that reduce spatial bias include center-left/down-left, center/down-center, down-left, and up-center/up-right. These positions guarantee that the AI model is trained to recognise TOIs across a range of visual placements.

Different visibility levels are captured by the TOI's Pictorial Distance, which ranges from recognisable distances to instances that are utterly unrecognisable. By addressing important issues with perception robustness, this distance diversity guarantees that the AI system learns to detect objects at different proximities.

Last but not least, the TOI's 3D Orientation details the various perspectives from which the aircraft can be viewed, including the front, rear, down-right, right, and unknown orientations. This approach improves the AI model's capacity to generalise under various real-world viewing circumstances by encompassing a variety of viewpoints.

The output CuneiForm Abstract Image is as below:

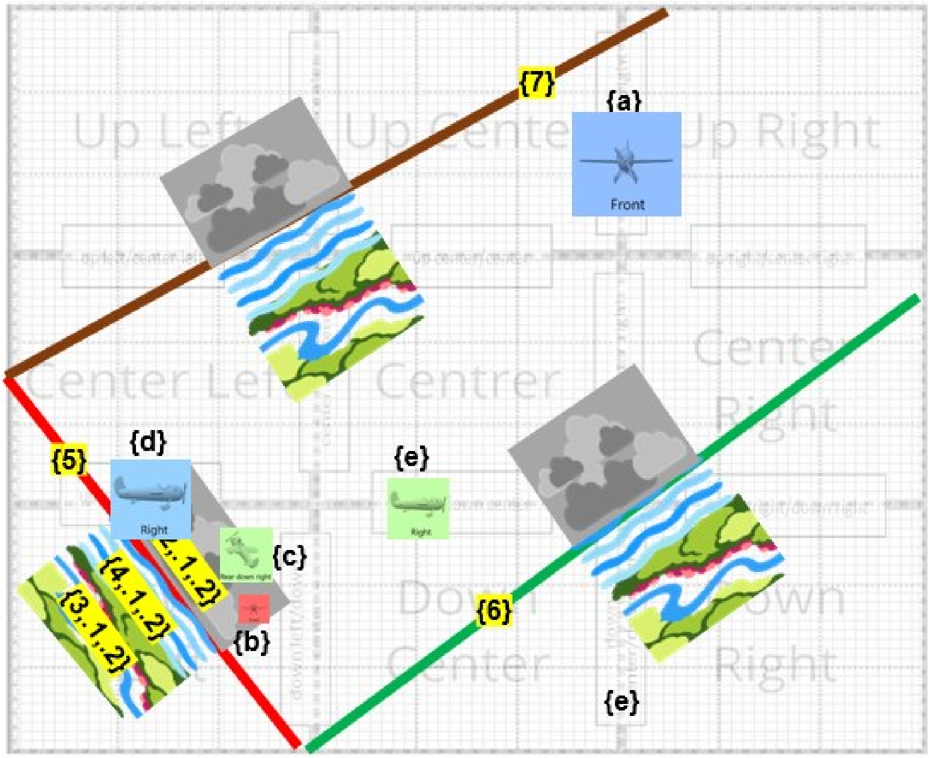


Figure I.33 Example CuneiForm for AVOIDDS. Coloured straight lines represent the horizon attitudes covered. Colouring of the TOIs orientation represents the pictorial distance.

Figure I.33 illustrates an Example CuneiForm representation for the AVOID dataset, showcasing a structured approach to characterising training data using pictorial positioning, TOI orientations, horizon attitudes, and environmental context. The figure employs coloured straight lines to indicate the visible horizon attitudes covered within the dataset. At the same time, the TOIs (Targets of Interest) are colour-coded to reflect their respective pictorial distance categories.

The numbered annotations on the horizon attitudes reveal information about the observed environment's spatial inclination. The Negatively Tilted Lowered Horizon 5 is represented by the red line, the Positively Tilted Lowered Horizon 6 by the green line, and the Negatively Tilted Elevated Horizon 7 by the brown line. These horizon attitudes show that a variety of viewing conditions are included in the dataset, simulating actual differences in aerial perspectives.

The aircraft instances in the figure represent the TOIs (Targets of Interest), each of which has a unique spatial placement and orientation. For example, the TOI a shows a single-engine propeller aeroplane 1 at the up center/up right 1.7 position, photographed from the front 1.12 angle. Additional diversity of viewpoints is provided by the TOI marked {d}, which is positioned at

Right {1.14}. A perception model can generalise across various viewing conditions thanks to the enriched training set created by these variations in 3D orientations.

Furthermore, the colour coding of the TOIs indicates their pictorial distances. To ensure variability in how objects appear within various observational contexts, the dataset includes examples of recognisable TOI distances {1.9}, clear close TOI distances {1.10}, and extremely unrecognisable TOI distances {1.11}.

Furthermore, the background objects within the dataset include stratus clouds {2}, green terrain {3}, and water surfaces {4}, reinforcing the need for AI models to operate reliably under different environmental conditions. These background elements maintain static motion {2.1, 3.1, 4.1}, and the dataset ensures that there is no motion blur {2.2, 3.2, 4.2}, thus preserving the clarity and stability of the TOI within the training images.

The figure below shows how the Cuneiform is abstracted. We associated the instantiated images with the part of the cuneiform that abstracts it. For example, image 50.jpg initiates the green horizon line.

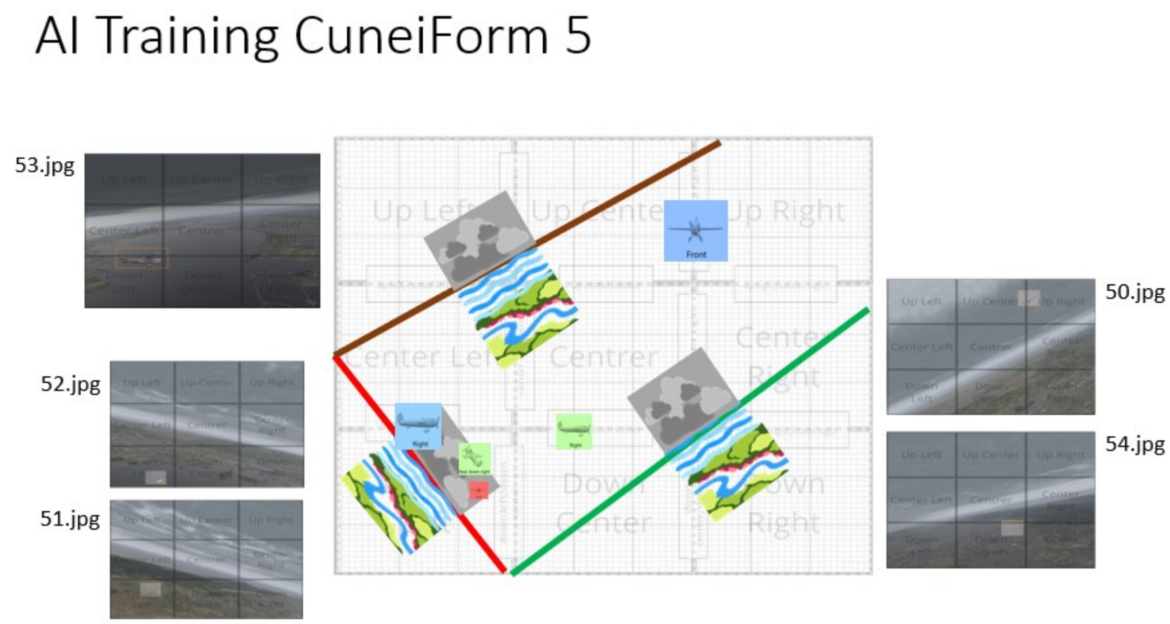
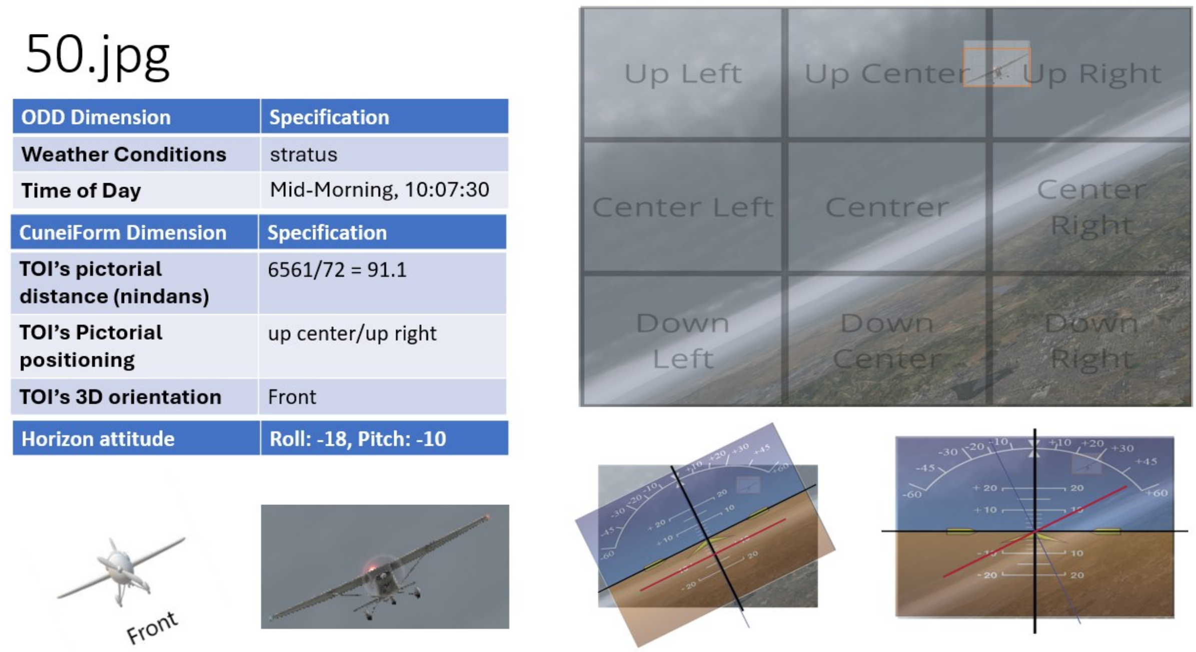


Figure I.34 How cuneiform 5 is instantiated in the AVOID training dataset sample

Figure I.34 suggests that there should be other similar training examples in the training dataset itself that they would have used. However, the team did not provide the actual 72,000 images they generated, and it appears as if the team had done a generate_traffic_data.py run and generated the sample images in the repository. For the sake of this case study, as we said at the beginning, we are assuming that the dataset is a sound statistical representation of a dataset generated for the purpose of the AVOID system.

I.10.4 CuneiForm Validation Artifact

The CuneiForm Validation Artifact serves as a structured mechanism to assess the epistemic coverage and trustworthiness of datasets used in AI-based safety-critical applications. This artefact provides concrete evidence supporting the dataset's adequacy in training machine learning models by mapping various perceptual dimensions, such as pictorial distance, 3D orientation, time of day, and horizon attitude, within the Operational Design Domain (ODD**)**. By incorporating detailed specification tables, annotated images, and computational analysis tools, the CuneiForm artefact enables a rigorous validation framework that can be used to support safety case claims regarding the dataset's sufficiency for real-world operational scenarios. In this section, we will describe the artefact using an example taken from AVOIDDS validation report for one of the training images samples in Figure I.32:


50.jpg

ODD Dimension	Specification
Weather Conditions	stratus
Time of Day	Mid-Morning, 10:07:30
CuneiForm Dimension	**Specification**
TOI's pictorial distance (nindans)	6561/72 = 91.1
TOI's Pictorial positioning	up center/up right
TOI's 3D orientation	Front
Horizon attitude	**Roll: -18, Pitch: -10**

Figure I.35 Example CuneiForm validation artefact for 50.jpg sample image

I.10.4.1 Structure of the CuneiForm Validation Artifact

Each CuneiForm Validation Artifact consists of multiple structured components designed to capture and evaluate the dataset's coverage over essential perception dimensions. The artifact includes:

1. CuneiForm and ODD Specification Table
   - This table systematically classifies the dataset's attributes concerning its Operational Design Domain (ODD) and CuneiForm dimensions.

- It includes information on weather conditions, time of day, pictorial distance, pictorial positioning, 3D orientation, and horizon attitude of the Target Object of Interest (TOI) in the dataset.
- An example is shown in the validation artifact for the training image "50.jpg", where the conditions are specified in the following training classes as:
    - Weather Condition: Stratus
    - Time of Day: Mid-Morning, 10:07:30
    - TOI Pictorial Distance: 6561/72 = 91.1 (measured in nindans)
    - TOI Pictorial Positioning: Up Center / Up Right
    - TOI 3D Orientation: Front
    - Horizon Attitude: Roll: -18, Pitch: -10

For example:

Table I.37 ODD and CuneiForm training classes

ODD Dimension	Training class spec
Weather Conditions	clear
Time of Day	late afternoon, 19:20:53
CuneiForm Dimension	**Training class spec**
TOI's pictorial distance (nindans)	6561/15 = 437.4
TOI's Pictorial positioning	down right
TOI's 3D orientation	front down right
Horizon attitude	**Roll: 1, Pitch: 7**

2. Graphical Evidence for Validation
    - The CuneiForm Canvas:
        - A 9x9 grid overlay on the image highlights the TOI's pictorial positioning within the image frame.
        - In the case of 50.jpg, the TOI is positioned in the upper center/right region, visually demonstrating its placement within the dataset.
    - Pictorial Visible Horizon Attitude Indicator (PHI) Tool:
        - The PHI tool measures the horizon attitude by computing the scene's roll and pitch angles relative to a level horizon.

        - For 50.jpg, the PHI tool indicates a roll angle of -18 degrees and a pitch of -10 degrees, which is crucial for evaluating model robustness under various horizon distortions.
    - 3D Orientation Abstraction:
        - A visualised TOI extraction showing the aircraft's 3D orientation class is included to provide an abstract classification of the object's orientation.
        - In 50.jpg, the TOI's orientation is front-facing, meaning the model will need to recognise the aircraft's frontal view under the given environmental conditions.

3. Incorporation into the Safety Case
    - The CuneiForm Validation Artefact is critical for safety case development, as it provides quantifiable and reproducible evidence regarding the dataset's epistemic coverage.
    - In machine learning-based object detection systems, the combination of graphical evidence and structured tabular specifications reduces epistemic uncertainty by enabling traceable validation of the dataset's capacity to generalise across various conditions.
    - Diverse pictorial positioning, horizon attitudes, and environmental contexts are included to guarantee that AI-based safety-critical perception systems are assessed against actual operational difficulties rather than merely hypothetical situations.

We assess the dataset's compliance with the ALARP standard for epistemic uncertainty reduction using the CuneiForm Validation Artefact. This methodology improves the dataset's credibility in supporting AI-driven perception models by utilising grid-based positioning, PHI horizon alignment, structured CuneiForm dimensions, and 3D orientation abstractions. The safety case claims are directly supported by this systematic validation procedure, which also strengthens the dataset's dependability in identifying and categorising objects in real-world settings, including Black Swan situations.