SYNOPTIX

systems thinking

AI ASSURANCE IN DEFENCE: CHALLENGES IN OPERATIONALISING JSP 936

December 2025

TABLE OF CONTENTS

1 AI Assurance in Defence: Challenges in Operationalising JSP 936

1.1 Introduction

This short report explores a set of challenging requirements drawn from JSP 936 Part 1, Version 1 ("JSP 936"). These requirements have been clustered into eight thematic challenge areas, each representing a distinct aspect of the broader implementation landscape. For each area, the report poses a series of questions and, where appropriate, offers brief reflections on the underlying issues.

JSP 936 is a well-structured and valuable document. Its purpose and intent are not in question. However, the real complexity lies in translating its requirements into operational practice. The challenges of operationalisation are not only multi-faceted and deeply interconnected, but also inherently difficult to resolve. This report aims to surface and explain some of these challenges, to inspire further development and support wider awareness of how these challenges may impact the Ambitious, Safe, Responsible deployment of AI across UK Defence.

1.2 Development Process

Initial synthesis of requirements embedded within JSP 936 Part 1 revealed approximately 272 requirements (121 "should", 151 "must") that make up this directive. A number of requirements would require further refinement to meet requirements quality standards, so the number of atomic requirements contained within this set would be likely to increase, but this serves as an initial benchmark.

A manual review was undertaken of the requirements, where each requirement was categorised by difficulty. Approximately 35 were identified in the "significant" category: those that represent a significant challenge with current maturity of AI technology or techniques for undertaking AI assurance. These were clustered into 8 key challenges by identifying these and trends. Whilst this is a subjective interpretation, it does meet this report's intended purpose. It is not intended to be a complete record of every challenge with implementing JSP 936, simply to highlight some key issues that this might bring.

Challenges with Implementing JSP 936			
Justifying Adequacy of Evidence and Argument	Managing Human Interaction with AI	Defining the Operational Environment	Dealing with AI as a System of Systems Component
Assessing and Maintaining AI Performance	Analysing Safety and Security in AI-enabled systems	Measuring Ethicality	Mitigating the Inherent Complexities of AI

This analysis was purely based on JSP 926 Part 1, version 1. It is important to note that JSP 936 Part 2 and the AI Practitioner's Handbook may provide best practice, guidance, and methods to mitigate some of these challenges.

2 CHALLENGES

For each challenge, the relevant requirements from JSP 936 are identified (including a link to the Directive paragraph in question), and a series of questions and sub-questions are posed. Where relevant, there is a brief discussion around some of the issues or challenges contained within each question.

2.1 JUSTIFYING ADEQUACY OF EVIDENCE AND ARGUMENT

2.1.1 Requirements from JSP 936

"Risk Owners must judge that the evidence supporting confidence in the system is adequate throughout the AI lifecycle" (P. 6)

"Data should be demonstrated as correct" (P.157)

"Where High-Level Requirement behaviours are to be implemented through AI and are not directly decomposable into Low-Level Requirement, the combination of the training algorithm and data requirements must be demonstrated as meeting the intent of the High-Level Requirement" (P.144)

2.1.2 Judgement of Adequacy

How do we judge that evidence is adequate?

Does adequacy change in different contexts?

Assurance primarily exists as a mechanism to demonstrate compliance with requirements, to support acceptance by stakeholders that their requirements have been met, or to support certification against applicable regulation of standards. Adequacy, therefore, supports the demonstration of justified trust, where those accountable for systems are attempting to demonstrate evidence that they have sufficiently addressed information asymmetries by measuring, evaluating, and communicating reliable evidence. For AI systems, where they operate in a complex socio-technical space, this means that there are a significant number of dimensions of adequacy:

- Legal adequacy: do we have a robust and legally defensible argument against possible and reasonable civil and criminal action? Are we content that the remaining legal risk has been suitably mitigated?
- Ethical adequacy: are we operating within the general ethical consensus of civil society, and the framework of reasonable ethical consideration - bearing in mind that ethical acceptability is highly subjective and variable? As well as operating ethically, are we are being seen to be ethical? Would a reasonable outside observer be able to sit in on our assurance review meetings and be content that we're considering and thinking about ethical issues as part of our development?
- Technical and safety adequacy: do we have sufficient evidence that the system (in the real world) performs as expected? Do we have sufficient evidence that the hazards possible within the system are well controlled?

These cannot be considered in isolation from each other - a technical inadequacy could lead to an ethical inadequacy which could then lead to a legal inadequacy, for example. They also can't purely consider an AI model - they need to integrate perspectives around a sociotechnical AI system deployment.

2.1.3 Adequacy of Data

How can data be "correct" – given that data representing a real-world system will only ever be an abstraction?

Beyond a small set of trivial "real world" problems, it is impossible for data about a real-world system to be correct. This is especially true in the situations which AI systems are precisely suited to deliver the greatest value - where the real world is complex and uncertain. The dimensionality of the real world is continuous and infinite - so any data model will only ever be an abstraction or representation. Rather than being correct, what we really require is that:

> *The dataset created out of the real-world system must be sufficiently representative of the complexity present so that an AI system trained from it would be able to identify all relevant features present in the real world*.

This is still entirely impractical, but at least it isn't impossible.

2.1.4 Requirements for AI

How do we map requirements to AI system design?

How do we determine adequacy against "intent" of high-level requirements?

AI systems aren't typically designed with formal system requirements. The standard approaches for AI development typically utilise Agile delivery frameworks and focus on user stories and backlog items to plan work. Whilst user stories are requirements - of sorts - the gap in verifiable system level requirements can cause challenges for conventional acquisition programmes. One reason why this is the case is that AI systems are also typically designed inductively - they work from the data - rather than deductive - working clearly from requirements. When AI is integrated into a larger system, particularly when the system isn't a pure software system, requirements are clearly going to be more important.

In addition, requirements engineering needs may be different depending on the type of AI system involved. For example, Natural Language Processing (NLP) has a very different set of needs than Computer Vision (CV). For example, needs may vary due to:

- Data Characteristics:
    - CV: images, video streams; requirements for resolution, frame rate, lighting, etc,
    - NLP: text, audio, (images); requirements for language, dialects, context sensitivity, etc.
- Model Behaviour
    - CV: requirements for different physical locations, variety in detected objects, etc,
    - NLP: requirements for semantics, cultural interpretation, translation, etc.
- Integration Context:

  - CV: requirements for integration to cameras/sensors, leading to latency, bandwidth, and environmental requirements,
  - NLP: requirements for integration to document processing systems, leading response time or privacy requirements.

Whilst this set of requirements is in no way complete, it demonstrates the variance in needs from different types of AI systems, and the corresponding difficulty in defining consistent requirements.

2.1.4.1 Key Challenges in applying Requirements Engineering to Machine Learning:

Stakeholder Expectations:

- Stakeholders often have high expectations for AI systems but struggle to understand their true capabilities and limitations. This gap can lead to unrealistic demands and misaligned goals.

Defining Requirements:

- Writing precise specifications for data-driven features is challenging, especially when concepts like “a person” need machine-comprehensible definitions.
- Requirements are difficult to draft when the necessary data is not yet available, creating uncertainty in early stages.
- Legal and ethical considerations add further complexity to defining requirements for AI systems.
- Many non-functional requirements are harder to define for AI systems (e.g. performance, reliability, robustness) and can also be very challenging to verify. In addition, these requirements can easily lead to optimization for the wrong objectives. Ensuring alignment with intended outcomes is a persistent challenge.

Nature of AI:

- Many AI systems are non-deterministic or probabilistic, making verification of requirements extremely difficult.
- Emergent behaviours cannot be fully anticipated or specified in advance, adding unpredictability to system design.

Trade-offs and Optimisation Factors:

- Balancing competing non-functional requirements is complex; such as privacy versus transparency or fairness versus accuracy.
- Expressing these trade-offs in requirements and deciding which compromises are acceptable, and at what cost, remains a major hurdle.

Skillsets and Experiences:

- Data scientists or AI engineers typically lack deep expertise in formal requirements engineering, while requirements engineers often have limited experience with AI systems.
- The absence of established guidelines in this area compounds the difficulty, leaving teams without clear best practices.

2.2 MANAGING HUMAN INTERACTION WITH AI

2.2.1 Requirements from JSP 936

"The fact that higher levels of autonomy typically reduce the potential for human decision-making must be considered when applying these requirements" (P.35)

"An assessment must be made of the impact that AI could have on the identified human stakeholder groups" (P.52)

"An analysis of the allocation of functions between human and AI agents and AI behaviours across all modes of function and levels of autonomy must be conducted" (P.125)

"Training Needs Analysis for users of AI-based systems should allow them to calibrate their trust in the system under different use cases and conditions" (P.130)

"In cases where AI is supporting important/risky decision-making, its output should have additional checks applied that are undertaken by a relevant subject matter expert" (P.180)

2.2.2 Levels of Autonomy

How does higher levels of autonomy interact with other requirements? Does it "raise the bar" to consider adequacy of confidence?

How do you manage defining interactions at different levels of autonomy? Which ones are common between different levels, and which ones change?

The first question, prior to discussing the level of assurance required at each autonomy level, is defining the levels themselves. Key dimensions include decision-making authority, execution of actions, and handling of exceptions or fallback scenarios. These dimensions range from full human control (manual operation) to complete system autonomy (unsupervised operation). Some examples of levels include:

- Lloyd's Register (2017) outlines a progression from manual control (AL0) to un-supervised autonomy (AL6), emphasising increasing independence in decision/action.
- SAE J3016 (2021) adds granularity by distinguishing between execution, environment monitoring, and fallback performance across driving modes.
- Parasuraman et al. (2000) propose a scale of human-computer interaction, from full human control to full computer autonomy, applicable across decision/action phases.
- ISO 23860 (2022) introduces a matrix of control and automation degrees, identifying configurations such as Fully Autonomous (FA), Autonomous Control (AC), Operator-Automation (OA), and Operator Exclusive (OE), based on the balance of human and system responsibilities.

Together, these frameworks highlight autonomy as a multidimensional construct involving control distribution, system capability, and operational context. They also highlight the fragmentation of solutions to this challenging question, which is fundamental to assuring autonomy.

The level of assurance required for autonomy depends on multiple interrelated factors that influence safety, reliability, and human-system interaction. These include the system's capacity to affect the environment, the degree and effectiveness of human monitoring, and the ability to intervene during error states. Assurance must account for the risk inherent in the operational context, the type of task, and the automation's role across the decision chain (information acquisition, analysis, decision-making, and action execution). Human performance considerations -such as mental workload, situational awareness, complacency, skill degradation, and trust – directly impact assurance needs, as do automation reliability metrics (accuracy, variance, MTTF) and the challenges of communicating these statistically. Finally, the consequences and costs of incorrect decisions or actions determine the rigor of verification, validation, and monitoring required to certify autonomy at a given level.

Reusing assurance evidence for autonomy is challenging due to changing component criticality as autonomy levels increase, which can shift safety responsibilities across the system. Managing mixed-criticality architectures adds complexity, requiring clear segregation and assurance strategies. Furthermore, the definitions, assumptions, and limitations underlying previous assurance arguments must be explicitly articulated to ensure transparency.

Relevance assessment is also essential - evidence must be validated against the new operational context and autonomy level to confirm that prior arguments remain applicable and do not introduce hidden risks. In addition to the validation at different levels of autonomy, additional effort must be introduced here to analyse reversionary modes. In addition to the validation required due to the operation in that mode itself, validation would also be required for the entry/exit from that mode - the state transition itself.

2.2.3 Impact Assessments

How do you avoid impact assessments becoming cumbersome "tickbox" exercises?

There is limited information available in the literature about the effectiveness of AI Impact Assessments. Although they are mandated in many AI management standards and regulation (including ISO 42001, EU AI Act, etc), research identifies a number of difficulties with conducting these assessments in ways that actually identify and mitigate unwanted ethical and societal harms. As Stahl et al (2023) identifies, all AI systems will have some impact to some extent - as this is precisely the useful quality that we're implementing an AI system to create. This means that we need to:

- define and quantify impact in ways that allow us to measurably identify what impact matters, and also that the measures of impact are those that are consequential in the outputs and outcomes created by the system.
- develop systematic ways to develop cost-benefit, performance-impact, risk-reward, or other trade-offs between different system architectures, designs, realisations, or implementations.
- manage concerns between individuals, groups of individuals, societies, organisations, governments, market forces, national states and geopolitics, and international considerations.
- develop ways and mechanisms for ensuring that impact assessments stay up to date with system capabilities and use cases, so that assessments stay up to date.

- develop ways and mechanisms for assessing the impact of "undesigned" capabilities of the system (particularly with more general-purpose AI systems).
- understand how can we assure the content of these assessments? How are they subject to external or objective scrutiny, or are they a purely internal exercise focussed on provision of internal assurance for decision making?
- understand how do we operationalise and integrate these impact assessments into the actual decisions that matter - both high-level usage and deployment decisions, but also low-level technical design and architecture decisions where significant change to the system can occur?

2.2.4 Confidence and Calibration

How can we define the confidence that someone has in an AI system? How do we understand the basis for this confidence – understanding which factors influence decision?

Is it a problem if users are "calibrated" against "wrong" factors?

Trust is an attitude that is relevant to automation in situations that include:

- levels of uncertainty,
- a cooperative relationship between at least two entities,
- some exchange between the two entities.

If there is no uncertainty (and therefore risk), trust is not required - outcomes are either certain or irrelevant. Given that there is uncertainty as a key part of this, this implies that there must be the possibility of wrongfully trusting or wrongfully distrusting. This leads to the idea of calibrated trust - describing how well trust matches the true capabilities of the automation. Trust can be affected by a variety of different components; some examples are below:

System-related components	Human-related components	Context-related components
Performance-based	**Ability-based**	**Tasking-related**
• Dependability • Performance • Predictability • Reliability	• Competency • Expectancy • Expertise • Prior Experience • Workload	• Risk • Task complexity • Task type
Attribute-based	**Characteristic-based**	**Teaming-related**
• AI Personality • Anthropomorphism • Appearance • Behaviour • Communication • Level of Automation • Reputation • Transparency	• Attitudes toward AI • Comfort with AI • Culture • Education • Personality traits • Propensity to trust • Satisfaction	• Communication • Interaction frequency • Shared mental models • Tenure

Trust calibration can be understood across several dimensions. At its simplest, calibration is the balance between justification and trust - whether the human's perception of the machine capabilities is reflective of the true, "ground zero", capabilities. There are also additional complexities, such as:

- Exogenous vs. endogenous calibration distinguishes whether trust is adjusted before/after interaction (exo) or during interaction through interventions (endo).
- Warranted vs. unwarranted calibration addresses whether trust accurately reflects system reliability or is influenced by factors like reputation or anthropomorphism.
- Static vs. adaptive calibration considers whether trust adjustments remain fixed or dynamically adapt to user needs and behaviour over time.
- Performance-oriented vs. process-oriented calibration differentiates between providing reliability metrics and explaining system processes, the latter requiring users to interpret how process details relate to performance.

All of these factors combine to result in a complex, delicate balance, which requires careful management and engineering to avoid unintended consequences.

2.2.5 Expertise, Oversight, and AI

If the human needs to be able to understand the system's outputs, what information do they need to see? How can we determine which of the inputs to the system were important, and need to be provided to the user?

What is the minimum effective level of information that we need to provide to the operator to allow them to gain adequate situational awareness?

How do we design effective oversight and interactions mechanisms, so that operators are able to effectively interact with and oversee systems?

How could the human-machine oversight interaction breakdown and fail? What might cause it to fail, and what would the consequences be? How can we design systems to be resilient to oversight failures?

What does an SME for an advanced AI system look like? Does expertise in the performance of traditional systems translate to expertise in the performance of advanced AI systems (particularly when integrated elements of autonomy)?

2.2.5.1 Transparency and Situational Awareness

We can't provide an overseeing operator (which could be a subject matter expert but may not be in many situations) with all information about the system, its environment, and its decisions. If we could (in a way that an operator would effectively understand), why would we be adding AI into the system? Therefore, we have to produce a meaningful subset of information, and in particular the minimum effective level of information that allows the operator to gain adequate situational awareness.

This will, of course, be context dependent, but 3 key questions of situational awareness generally can be:

- What's going on, and what is the system trying to achieve?
    - Key elements: purpose, goals, process, intentions, progress, performance
- Why does the system do it?
    - Key elements: reasoning process, belief, purpose, constraints (environmental and otherwise)
- What should the operator expect to happen?
    - Key elements: potential limitations, uncertainty, likelihood, history of performance, projection to future/end state

2.2.5.2 Human-Machine Interactions and Oversight

Particularly when interacting with autonomous systems, human-machine interaction is a complex subject. Current debates typically try and reduce these interactions to simple forms (e.g. human-in-the-loop vs. human-on-the-loop vs. human-out-of-the-loop). These don't capture the complexity and diversity of modalities by which humans can interact with complex systems.

Particular focus also needs to be applied to both the objectives of AI use/autonomy, as well as the risk factors of the system and application. For example, if we are aiming to achieve speed of processing, the oversight mechanisms might look very different to if we are aiming to achieve compliance or quality objectives. Given the complex nature of these systems, it's also likely that there are multiple system objectives that need to be balanced, or even objectives that vary over different sub-components of the system. Equally, core risk factors of the application (like the severity of consequences for "incorrect" decisions, the time sensitivity uncertainty of the decisions, and the reversibility of an "incorrect" decision once it has been made) will all affect the needs, objectives, and appropriateness of oversight and control mechanisms.

2.2.5.3 Current and Future Skill States

Integrating AI into existing applications may fundamentally change the way that those applications and contexts work. Indeed, this might well be an optimal situation, as it could mean that system design has truly worked to optimise the system towards the strengths of the AI model, rather than just implementing AI for the sake of this.

However, this does pose an interesting question in the use of operator or SME expertise in evaluating the outputs of a system. For a sufficiently "revolutionary" AI system, would the change to the way the system works invalidate or decrease the expertise of those who were previously experts in conventional systems? How transferrable is this domain expertise, and how can we tell when we have reached the limit of someone's knowledge? For example, someone who is expert in evaluating and cohering intelligence outputs from human analysts may not have adequate understanding of the failure modes of Large Language Models to understand the limitations of their use in processing open-source intelligence information.

2.3 Operational Design Domains/Operational Environment

2.3.1 Requirements from JSP 936

"For a civilian RAS being used in a Defence context, the military delta in the Operational Design Domain must be identified" (P.36)

"The ODD for the AI should be identified" (P.38)

"The operating context for the AI components must be clearly defined and communicated to relevant stakeholders" (P.75)

"Appropriate response to reasonably expected inputs outside of the intended design must be defined and demonstrated" (P.78)

"The data should be a sufficiently accurate reflection of the real-world application" (P.157)

"Data should be demonstrated as correct" (P.157)

"AI Assurance must include assurance of behaviour where excursions from the AI ODD may reasonably be expected to occur" (P.209)

2.3.2 Defining the Operational Environment

How do we determine that any definition of the operational environment is sufficiently accurate in describing the real-world operational envelope – for anything sufficiently complex?

Identifying elements of the ODD is easy – which bits of it matter? When will changes to parts of the ODD change the way that the system behaves?

Some aspects of the "military delta" will be obvious – but how can we identify if we have captured all aspects of it?

Operational Design Domains (ODDs) are traditionally derived from a human-aligned view of the world, but does such an approach work for AI systems? Humans intuitively understand variations and expected limitations in their environment, and our assumptions about what information is required to represent the complexity of the real world are not necessarily that which are required by AI systems to construct a machine world model. Humans also tend to anthropomorphise machine behaviours, often leading to assuming that AI system's behaviours represent higher levels of understanding that they actually do - it's not uncommon for sophisticated combinations of Skills and Rules to imitate higher-level Knowledge and Reasoning understanding modes, without the AI system undertaking the actual behaviour represented by a human performing at these levels.

Generally, the purpose of an ODD is to seek to define the operational boundaries within which a system is expected to function safely and effectively. However, its purpose needs to extend beyond static definition, in order to capture representative real-world limitations. It must address how we identify gaps, manage uncertainty, and anticipate hazards that arise when assumptions fail.

Managing change throughout the system lifecycle is central to an ODD's robustness. Real-world conditions evolve, introducing temporary exceptions or permanent drifts that challenge prior assumptions. Similarly, model capabilities shift through retraining and performance updates, creating a dynamic interplay between environment and system competence. ODDs often represent the world as previously experienced, yet this approach falters in the face of black swan events: rare, high-impact scenarios outside historical data. Should the system's ability to perceive and respond to novel phenomena become an explicit component of an ODD? This question underscores the need to integrate capability modelling into domain definition.

Sensor limitations further complicate the picture. What a system can "see" directly influences its operational safety. The October 2023 Cruise incident, where a vehicle dragged a person beneath it, illustrates this vividly: the system lacked any representation of such a scenario because its sensors and training data provided no basis for detection. Therefore, an ODD must incorporate sensor constraints and their real-world implications, ensuring that environmental perception aligns with operational expectations.

2.3.2.1 Defining an Operational Design Domain

Consider, for example, a 4-dimension breakdown, as is sometimes adopted in "gold-standard" autonomous vehicle ODDs. Initially, definition is required of the Operational Environment: perhaps including factors such as location, infrastructure characteristics, or communication methods. This grows increasingly complex as the "rules" that govern the environment become more complex – such as a military technology deployed in the chaotic environment of a warzone. Additional complexity also rears its head when you consider the virtual and social aspects that feature in every modern system: information architectures, system interfaces, human-machine interfaces, political and geopolitical trends, and social and cultural values.

Add to this complexity, then, the Event Detection and Response characteristics – the system's ability to perceive that in the world around it. Key here is the detection of all relevant entities – those that matter to the system, as well as management of false positives and false negatives. As well as detecting entities themselves, behaviour must also be determined – what is the expected behaviour of these entities in this environment? Combine this with the Active System Behaviour and Decisions: the system's ability to effect change in its environment, such as the set of possible operational outcomes, goal setting and goal seeking behaviours, operational modes and mode transitions.

Finally, we must also consider the Failure Modes and Fault Management Parameters. These separate into 3 general categories: System Limitation (inherent boundaries or constraints that affect a system's capabilities or performance - not the result of a failure or fault with the system, but rather built in by design or component choice); System Fault (a fault, error, or malfunction in the system such that the system is unable to effectively perform its function); and Fault Responses (aspects of a system-level view of fault detection and mitigation that mitigate system faults).

Importantly, these are not independent characteristics. They are highly integrated and interdependent on each other. For example, the Event Detection and Response characteristics may well be highly dependent on certain operational modes or mode transitions within Active System Behaviour and Decisions, as the system overheats and processing capabilities shut down (a System Fault from Failure Modes and Fault Management Parameters), leading to a failure to determine all relevant entities in the Operational Environment.

2.3.2.2 Managing the Military Delta

There are two main aspects to this: the military delta due to the different requirements or circumstances due to the military operational needs (e.g. operational environment, platform integration, security considerations, etc), and the military delta due to procurement mechanisms, requirements, or standards.

The real challenge isn't in identifying these requirements (though this can be difficult!), it is - much like many other challenges - understanding if you have captured all of the relevant requirements. These missed requirements (the unknown-unknowns) are where risk is likely to sit in the adaption of civilian-designed technologies for the military operating environment.

2.3.3 Use of the System

How do we cover the usage of the system outside the intended ODD? How can we detect that operation outside the ODD is occurring?

Is analysing misuse, abuse, and disuse of the system sufficient, or do you need more?

When we define the boundaries of our ODD, we are implicitly setting out the operational space that we expect our system to be used in. However, there are two critically important areas of this space:

- reaching the edge of this boundary - where you are "operating under boundary conditions",
- over the edge of this boundary - operating in a space beyond the design.

Where this can be particularly challenging is where the boundary is very jagged - i.e. it is not a smooth surface of capability, but rather varies significantly between tasks, actions, or context's that a human operator would consider to be very similar. This is behaviour exhibited, for example, by current-generation Large Language Models (e.g. able to develop complex mathematical equations but can make errors with the date that the work was completed).

A second issue is to reliably determine when the system is operating near to or outside of the ODD boundary. Assuming the ODD is sufficiently broad (e.g. in a general-purpose tool), this becomes a non-trivial problem. However, it is critically important to be able to have confidence in applied guardrails or control that are used to reduce the risk of system failures. In some situations, it may be a trivial exercise to detect some deviations of an ODD. For example, in a self-driving car where the ODD is primarily based on location, GPS tracking may provide strong indication of likely breach. However, many ODDs represent much more complex behavioural states, where there is not any definitive data that can be linearly mapped to an ODD.

Finally, there is the additional lens of intent to add to the complexity. This contrasts the intent of the system's designer against that of a human user, and may be summarised through a **'**use, misuse, abuse, disuse' framework. For the system analyst, this presents the question: "Is the human user intending to use the output of the AI system in the way the designer intended it to be used?" This addresses some of the key concerns of highly-autonomous AI-enabled systems – that they are easily used beyond their intended scope of design, and this is where significant failure can occur. It can make a meaningful difference to intentionally use the system beyond the ODD (abuse) rather than non-intentionally (misuse). However, this can be seen to apply an

overly simplistic lens to a complex problem of human behaviour. For behaviourally complex situations (for example, when operators are under extreme stress), are additional lenses need?

2.4 Dealing with AI as a System of Systems Component

2.4.1 Requirements for JSP 936

"The level of influence/consequences of AI outputs on overall digital system performance should be identified and incorporated into overall system risk analysis" (P.43)

"Where AI interacts with other systems, the AI behaviour should be understood in the system of systems context" (P.65)

"The failure analysis must consider the potential for inter-system emergent effects and cascaded failures" (P.82)

2.4.2 Impact of AI within a system of systems

How do we "trace" the impact of AI throughout a wider system of systems? If the end-user interfaces with a downstream system – how can they understand the dependencies that sit throughout the information/decision chain?

Managing AI within a System of Autonomous Systems (SoAS) introduces significant complexity due to the interplay of autonomy, interdependencies, and emergent behaviours. Each system operates with its own objectives and limited environmental awareness yet must cooperate to achieve SoAS-level goals. If we need to understand the "who/what/when/why/where" for decisions across the SoAS, we need traceability. This can't just, however, result in the communication of every model internal of every system of the SoAS – this would just overwhelm operators or those trying to understand the information, and would fail to deliver the true goal of operational explainability. We need to design SoAS to create shared situational awareness and scalable observability, not large quantities of largely meaningless and unintelligible data.

Independent changes in one system can trigger unpredictable emergent behaviours at the SoAS level, complicating governance and risk management. When applying autonomous systems which possess causal agency, this creates challenges in aligning diverse system perspectives, integrating capabilities, and maintaining control when managerial authority is distributed across multiple organizations. Leadership fragmentation, conflicting objectives, and unclear accountability further exacerbate coordination and decision-making.

AI and autonomy amplify these challenges by introducing opaque decision-making processes, data dependencies, and security vulnerabilities. Black-box models hinder explainability and trust, making verification and validation difficult – especially when emergent behaviours or failure modes like miscoordination, conflict, or collusion arise. These dependencies may be implicit (such as shared training data, common or diverse optimisation objectives) and not documented, reducing visibility and increasing brittleness.

Testing and assurance become resource-intensive due to evolutionary development and interdependencies, while adversarial risks and multi-agent security threats increase exposure to systemic failures. These dynamics create a fragile ecosystem where small perturbations,

network effects, or adaptive feedback loops can destabilize the entire SoAS, demanding robust governance, transparency, and resilience strategies.

These SoASs are particularly vulnerable to adversarial exploitation – where attackers take advantage of these vulnerabilities and instabilities to create these cascading failures. Not only are these hard to prevent – as shown by all of the challenges throughout this section – but they are exceptionally hard to recover from. Not only do they often require significant reword to re-establish these complex systems, but the nature of these complex causal-chain failures often leads to additional difficulties in uncovering evidence of failure, tracking root-causes and failure mode chains, as well as attributing accountability across the SoAS.

Finally, there is a very significant challenge in understanding what matters within a SoAS. Autonomous systems make decisions constantly – and these decisions vary in scale. When AI models are underpinning the decisions of autonomous systems, it creates increasing difficulty in identifying which of these decisions are consequential to the behaviour of the system overall. For example, small threshold adjustments, pre-processing or filtering steps, or data fusion techniques may be adjusted dynamically, and these may emerge to create systemic failures. These “microdecisions” are high frequency, high opacity, extremely context-sensitive, and may well additionally be non-deterministic. All of these complexities provide significant challenge to manage, particularly within the SoAS context.

2.4.3 Interfaces with existing and conventional systems

When many of the systems in the system of systems context are conventional systems, perhaps already existence, and perhaps developed in very different ways, how do we determine the impact that the AI system has on them and the outputs that they produce?

One of the most significant areas of risk in the design of systems in general is interfaces to external legacy systems. This presents a number of challenges for assurance, above and beyond that of purely assuring the new AI system alone:

- There are implicit assumptions encoded into almost every interface - e.g. determinism, timing, stability - that aren’t necessarily explicit in Interface Control Documents, and so compliance with the intent of previous interfaces is required along with compliance with documentation.
- Adding AI systems into a legacy system of systems can shift or alter the attack surfaces of the system as a whole, as well as adding additional attack surfaces of the AI system itself - so it’s not enough just to assure the security of the new AI system, consideration must be given to the assurance of the wider System of Systems as a whole.
- Iterative development of AI systems is challenging to assurance dynamically on its own, but even more so when integrated with conventional systems that have legacy static assurance evidence.

2.5 Assessing and Maintaining AI Performance

2.5.1 Requirements from JSP 936

"When the system has more than one mode of operation or level of autonomy, the impact analysis must be conducted for all modes and autonomy levels" (P.54)

"Analysis for potential effects of reasonable failure modes must be carried out" (P.82)

"For collaboratively trained human-AI teams, assurance of the overall behaviour for each team must be provided" (P.131)

"Where wider system requirements include the expectation that AI will be modified in-situ then planning should include how this will be controlled and assured for continuing confidence" (P.135)

"Performance requirements for the AI should be clearly stated alongside functional requirements" (P.140)

"The AI Architecture must be able to incorporate the intended behaviour whilst protecting against entry into failure modes identified during hazard analysis" (P.150)

"Safe behaviours of the AI when exposed to inputs that fall outside of the ODD must be demonstrated" (P.171)

2.5.2 Instance Specific AI Performance Assessment

Is assurance at each level of autonomy required or practical? What can be reused?

How do you determine what reuse of assurance is acceptable, and what is no longer true?

The same applies to the assurance of overall behaviour for every team – in order to practically carry this out, is the level of assurance going to provide sufficient mitigation?

What change is required to the team (either human or AI) before re-assurance is required?

Levels of autonomy are complex, and this necessitates clear demarcation between different operating modes or states. This would, however, support useful resilience testing and graceful degradation architectures.

There is an additional challenge when managing human-AI teams that evolve together: where the human evolves their behaviour, trust calibration, and cognitive biases, and the AI dynamically learns to match decisions made by "their human", it presents a unique challenge for assurance. This also integrates into the System of Systems problem - if you have multiple instances of "the same" human-AI team that have evolved differently, do they require individual assurance, and how do you assure the system as a whole?

2.5.3 Management of AI Performance

Why are AI performance requirements separate to system performance requirements? The model should be integrated into the system, not separate.

Specifying AI specific performance requirements can be unhelpful when the aim should be to focus on the overall system performance requirements. It's important to be clear that the priority needs to be measures of system outcome, and then AI model performance outputs should support achieving those outcomes.

2.5.4 Defining and Understanding AI Failure Modes

If the failure mode lies within the AI's internal 'black box,' how can you determine the point of entry where the failure mode originates?

How can we define and manage the functional interactions between human and AI? How can we understand the basis for those interactions, as well as how they can fail?

What is a "safe" behaviour? Safe behaviour of the model, of the system, or both?

Analysing AI failure modes is hard:

- Failures are often interrelated to each other: failures both sides of the human-machine interface can multiple and propagate throughout the system.
- Standard techniques (e.g. FTA/FMEA) don't capture all of the complexity: mapping complex data, training, algorithmic, or non-deterministic failures modes against causal surfaces is hard, and a multi-lens approach is required for success. Lenses (perspectives on failure) could include human intent, algorithmic inaccuracy, cognitive bias, human factors, system resilience, and many others. Each lens is individually incomplete but used together this brings a depth and rigour to failure analysis.
- AI integration means the failure surface (gap between cause and effect) separates: the cause of failures can be significantly "further away" through the chain of system behaviours than is typical in non-AI systems.
- Increased need to manage human-cyber-physical system failures: whilst many conventional systems have this need, human-cyber-physical interactions are critical to the success of many AI systems, and this, combined with the increased human-machine interaction brittleness that is often seen in AI systems, can lead to a significant diversification in failure modes.

In addition, we don't just have to consider model performance failures, we also must include system outcome failures (e.g. mission failing) which are difficult to integrate into standard analysis techniques.

2.6 Analysing Safety and Security in AI-Enabled Systems

2.6.1 Requirements from JSP 936

"The AI Design should minimise insofar as is reasonably practicable the adversarial attack surface" (P.81)

"A Hazard Analysis must be undertaken to identify hazards introduced through the use of AI" (P.141)

"AI-unique safety risks must be analysed and included in the relevant wider safety cases and software and system risk assessments" (P.191)

2.6.2 Integration of Safety and Security

Given existing safety and security processes are not particularly integrated, how do we manage this in the AI space where it is even more crucial?

Systems, security, and safety work are generally not particularly integrated at the developmental level on Defence procurement programmes - despite relatively good integration at the acquisition level. For AI systems, where they are deeply interlinked and interdependent, this poses an even greater risk than in conventional systems. In addition to the cultural barriers to adoption between difference disciplines, there is a tooling and process barrier as well: different modelling tools are often used by different teams, which don't necessarily integrate together, and different disciplines have different standards to be compliant to. Work to integrate these together (for example, in secure systems engineering, the NIST 800-160 v1 and v2 standards) is critical.

2.6.3 Additional Safety and Security Vulnerabilities of AI Systems

What vulnerabilities are present in AI systems, as opposed to conventional systems?

AI Systems still have the same security objectives, as in any other engineered system. Maintaining confidentiality, integrity, availability, authenticity, and accountability is still vital. The difference lies in the mechanisms by which threats to these objectives may exist. Standard threat modelling tools (like STRIDE) might fail to grasp complexity – overlaying multiple models (like LINDDUN, for example) might be required.

There are also entirely new categories of threats. Adversarial attacks (such as data poisoning, evasion attacks, or model extraction) can cripple models – but significant risks exist in "silent attacks": where model compromise is challenging or even impossible to detect.

Certain classes of models also have individual vulnerabilities, or may be affected in different ways by attack classes. Large Language Models, for example, might be particularly vulnerable to multi-agent attacks. Conformity Bias can lead to abnormalities not being highlighted or low-magnitude signals being amplified, or monoculture vulnerabilities can give lead to inherent behaviour patterns interacting to create a vulnerability.

Finally, the wide use of third party models hosted externally creates significant supply chain compromises. Building a dependable AI system can't just include cyber assurance of directly created AI systems, but also on the foundations on which those systems rely.

2.7 Measuring Ethicality

2.7.1 Requirements from JSP 936

"If an AI system presents unacceptable negative ethical risks, deployment/ development must be halted in a safe manner" (P.91)

"Defence must behave ethically" (P.95)

"Defence must be seen to be ethical" (P.95)

2.7.2 Subjectivity of Defence

Ethics are subjective, referring to ethicality situates in subjective context. How do we capture this subjectivity in objective reporting?

How do we benchmark and present ethical risks, even when ethics are subjective, and different stakeholders have different views on acceptability?

Ethical risk appetite is highly variable and situational - even within teams it will vary between individuals, and the variety in contextual considerations means that defining fixed guardrails is very challenging. Integration (from a military context) into Article 36 decisions also involves cross-discipline interactions between technical teams, leadership, frontline warfighters, and legal teams. Quantification of ethical harms is a significant challenge - the socio-technical metrics (like bias, collateral effects, or unacceptable harms) are immature, contested, and challenging to measure - especially when some metrics could be desirable - for example, there is always bias in decision-making tools, and partly this bias is indeed useful and essential to their operation! However, unacceptable or unknown bias can be highly problematic from both ethical and practical perspectives.

Finally, ethical acceptability (or ethicality) isn't just about objective ethical assessment. Political, social, and media considerations must play a part, as they heavily interact with the parliamentary and political scrutiny that is a key part of Defence's environment.

2.7.3 Ethicality as Part of Defence

If one part of the Defence enterprise is unethical, does this compromise the ethicality of Defence as a whole?

Whilst expecting individual projects or systems to be responsible for the whole of Defence being ethical is clearly unrealistic, it does raise a number of key questions about the integration of AI into a system of systems context. Not only does the individual system need to be evaluated for ethical acceptability, the wider system and system of systems that it is incorporated into needs to be evaluated for ethical acceptability in the context of the AI system's capabilities and autonomy.

This also has to be a lifecycle project - it's not enough to just be done at integration. Ethical By Design needs to be built into the system's design from Concept, to reduce risk of unacceptable emergent ethical risks, and reduce wasted time and effort.

2.8 Mitigating the Inherent Complexities of AI

2.8.1 Requirements for JSP 936

"All models should be transparent" (P.167)

"All models should include appropriate explanations of their output" (P.167)

"All models should provide measures of uncertainty that are understandable to the various stakeholders" (P.167)

2.8.2 Inherent Complexities

How do we address failures and weaknesses in AI systems where the nature of the systems themselves is essential to the failure of the system?

A number of requirements require solutions that are still open, unresolved questions:

- Model transparency: AI models, especially deep learning, lack interpretable internal representations, and can block causal mapping. Transparency tools are often post-hoc approximations, not guarantees of correctness, further limiting their usefulness for assurance evidence.
- Appropriate explanations: explanations must be faithful and understandable, but many explainable AI (XAI) methods trade fidelity for simplicity. Careful management of explanations is also required to support trust calibration - misleading or subjective explanations could impact perceived system capability inaccurately.
- Measures of uncertainty: relying on uncertainty measures from AI systems is problematic: common confidence measures often fail to represent both aleatoric and epistemic uncertainty. Other methods (e.g. Bayesian or ensemble) can increase computing times, especially when used in a system of systems context (with uncertainty compounding through decision chains).
- Non-deterministic behaviour: Learning systems produce variable outputs under identical inputs due to stochastic training or runtime randomness. Every update may therefore invalidate previous evidence, and emergent behaviours in multi-agent systems will amplify this uncertainty.
- Brittleness and robustness: AI models can often fail under distribution shift, or in extremes of training data (black swan events). Robustness metrics are immature and there is limited research about their real-world effectiveness. Brittleness also is challenging to manage in the context of trust calibration - unexpected failures can lead to disproportionate drops in trust.

In addition, these only refer to the outputs of models, not the outcomes of systems, which adds further complexity.

3 Conclusion

This report has set out a set of 8 key challenges and associated questions with the implementation and operationalisation of JSP 936 Part 1 into UK Defence. Our future work will focus on addressing these challenges, through structured assurance arguments, robust domain-agnostic playbooks, and analysis and intervention techniques to allow Synoptix to extend our comprehensive Defence-focussed assurance offering. Future publications will include reports discussing the solutions and mitigations to these challenges, targeted for specific aspects of Defence activity.

How Synoptix Can Help

Synoptix's AI Assurance offering provides defence organisations with a structured, evidence-driven framework for establishing justified trust in AI systems. It tackles critical challenges such as managing human-AI interaction, ensuring safety and security, and mitigating complexity across systems of systems. Through the development and evaluation of assurance cases, Synoptix enables transparent communication of risk, performance, and ethical considerations throughout the lifecycle, ensuring AI solutions remain robust, reliable, and aligned with operational needs.

In high-stakes defence environments, decisions must be defensible and transparent. Synoptix's methodology – holistic, proportionate, and vertically traceable – supports governance, accountability, and compliance across domains like resilience, morality, and technical suitability. This approach reduces operational risk while fostering stakeholder confidence, enabling responsible AI adoption that safeguards mission integrity, reduces wasted effort and rework, and societal values.

About Synoptix

Synoptix is a specialist systems, cyber, and technology engineering company based in the South West of the UK.

We provide systems, security, and technology capability to a wide range of industries across the UK, alongside our own line of technology products and services. We add value by tailoring and delivering packages of work, as well as supporting clients to build and develop their own capability.

We offer high-grade AI products, focussed on computer vision, autonomy, and edge computing. When combined with our Systems Engineering capability, this means our AI Assurance offering brings together a rare set of skillsets to offer a unique mechanism for high-stakes AI Assurance.

Through our expertise and experience in UK Defence, we are able to offer a well-rounded and highly capable sovereign capability, relied on by Tier 1 Primes, SMEs, and directly by the MOD alike.

Author: Callum Cockburn, Technical Innovation Manager

Reviewer and Approver: Sam Farrow, Technology Director

www.synoptix.co.uk