Agentic AI System Design Report

S. Palis

Udacity Master's Degree in AI Capstone Design of Autonomous and Semi-Autonomous Agentic Workflows

February 15, 2026

GitHub repository:

https://github.com/MinervaRose/design-of-agentic-workflows

Overview

This project implements a small-scale Aerospace Anomaly Triage Agent designed to support monitoring of simulated aerospace sensor incidents. The system classifies incidents as Normal, Suspicious, or Critical and recommends a bounded next action such as monitoring, requesting additional information, or escalating to a human operator. The agent is explicitly designed as a decision-support system rather than an autonomous controller. It does not execute real-world actions and is constrained by safeguards that prioritize safe fallback behavior under uncertainty.

The system follows an agentic workflow combining deterministic tools, limited memory, LLM-based reasoning, and evaluation-driven revision. This architecture aligns with modern tool-augmented agent patterns in which language models reason over structured evidence and act through controlled interfaces (Yao et al., 2023). The goal of the project is not to maximize autonomy, but to demonstrate how agent orchestration, safeguards, and transparency can produce reliable and auditable decisions.

System Architecture and Design

The agent consists of four core components: deterministic scoring tools, a limited memory layer, an LLM synthesis interface, and a rule-based evaluation and safeguard layer.

Incoming incident reports are first validated for schema completeness and plausibility. Deterministic diagnostic tools compute an anomaly score and flags indicating specific risk signals (e.g., packet loss or drift). A lightweight memory system stores recent sensor history, enabling context-aware decisions without requiring heavy infrastructure.

The LLM synthesizes a structured decision object from tool outputs. However, the model is not authoritative: its output is passed through an evaluation function enforcing explicit safety rules. If inconsistencies are detected, the agent enters a bounded revision loop that feeds evaluation feedback back into the model. This reasoning–action–evaluation cycle mirrors tool-augmented agent architectures described in the ReAct framework (Yao et al., 2023).

Safeguards override model output when necessary. Missing critical data, low confidence, or rule violations trigger escalation rather than speculative decisions. This safety-first design reflects risk-aware AI governance principles emphasizing controlled autonomy and transparent fallback behavior (NIST, 2023).

Decision Logic and Behavior

The agent's decision logic is grounded in deterministic evidence. The anomaly scoring tool produces a numeric risk value and categorical flags, which serve as the primary basis for classification. The LLM interprets this evidence to produce a human-readable rationale and a structured decision, but cannot bypass rule-based evaluation.

During execution, the agent demonstrates consistent behavior across representative scenarios. Low-risk incidents are classified as Normal and monitored. Intermediate anomalies are labeled Suspicious, prompting continued observation. High-risk multi-flag incidents escalate to Critical and trigger human review. When required fields are missing, the missing-data safeguard activates and the agent requests additional information instead of guessing.

The revision mechanism ensures that the model corrects itself when its output violates policy. Evaluation feedback is explicitly returned to the model, allowing bounded self-correction. This prevents runaway loops while maintaining the ability to refine decisions.

Safety, Reliability, and Transparency

Safety is implemented through layered constraints rather than relying on the model alone. Deterministic scoring reduces hallucination risk by grounding decisions in explicit rules. Evaluation functions enforce consistency between risk scores and classifications. A maximum revision budget prevents unbounded agent behavior. When uncertainty remains unresolved, the system escalates rather than overcommitting.

This design reflects a principle of constrained autonomy: the agent is allowed to reason, but within strict operational boundaries (NIST, 2023). Structured JSON outputs and trace logs make decisions inspectable and reproducible. Each decision includes a trace summary containing score, flags, revision count, and evaluation history, enabling post-hoc auditing.

The architecture follows modern function-calling and structured-output design practices for tool-using LLM systems (OpenAI, 2024), ensuring clear separation between reasoning and execution.

Observed Behavior and Limitations

Empirical testing reveals both strengths and limitations. The agent reliably escalates high-risk scenarios and triggers safeguards under missing data conditions, demonstrating robust uncertainty handling. The bounded revision loop successfully corrects inconsistent model outputs when evaluation rules are violated.

A notable limitation emerges in borderline cases. Because the scoring tool uses hard thresholds, values just below anomaly cutoffs produce a score of zero and a fully confident Normal classification. Human operators might interpret such scenarios as ambiguous rather than risk-free. This behavior highlights a limitation of the simplified scoring model rather than the orchestration architecture.

Future systems should replace hard thresholds with smoother risk functions, probabilistic scoring, or calibrated confidence models. These improvements would allow the agent to represent uncertainty more faithfully near decision boundaries.

Ethical and Responsible Use Considerations

Even in a simulated environment, decision-support agents raise ethical concerns related to autonomy, accountability, and overconfidence. A key risk is automation bias: users may over-trust agent outputs if uncertainty is not communicated clearly. The system mitigates this risk by exposing confidence values, providing human-readable rationales, and logging decision traces.

Another ethical concern is unsafe autonomy. Systems that guess under uncertainty can propagate silent errors. The agent explicitly refuses to guess when critical data is missing, instead

escalating to human oversight. This aligns with safety research emphasizing fail-safe behavior and human-in-the-loop governance (Amodei et al., 2016).

The architecture prioritizes transparency and bounded autonomy over raw efficiency. This reflects a design philosophy in which AI systems support human decision-makers rather than replace them.

Future Improvement

Several improvements could enhance robustness and realism. First, the deterministic scoring tool could be replaced with a probabilistic model calibrated on real sensor data. Second, confidence estimation could incorporate uncertainty modeling rather than fixed thresholds. Third, the memory system could expand to include long-term trend analysis. Finally, adversarial testing and stress simulations could evaluate behavior under extreme or noisy conditions.

These improvements would not fundamentally change the agent architecture, but would strengthen its reliability and applicability to real-world decision-support scenarios.

References

Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K., & Cao, Y. (2023). *ReAct: Synergizing reasoning and acting in language models*. International Conference on Learning Representations. https://doi.org/10.48550/arXiv.2210.03629

National Institute of Standards and Technology. (2023). *Artificial intelligence risk management framework (AI RMF 1.0)*. U.S. Department of Commerce. https://nvlpubs.nist.gov/nistpubs/ai/NIST.AI.100-1.pdf

OpenAI. (2024). *Function calling and structured outputs*. OpenAI Documentation. https://platform.openai.com/docs

Amodei, D., Olah, C., Steinhardt, J., Christiano, P., Schulman, J., & Mané, D. (2016). *Concrete problems in AI safety*. arXiv. https://arxiv.org/abs/1606.06565