Operation Design in the Fifth Industrial Revolution

A. Harel, ergolight@gmail.com, 2025

The Fifth Industrial Revolution (5IR) is an emerging concept built upon the Fourth Industrial Revolution (4IR). It reintroduces the human element into technology-driven transformation. While the 4IR focused on automation, AI, and efficiency, the 5IR emphasizes collaboration between humans and machines, the ethical use of technology, and sustainability (Kumar and Shankar, 2022).

This research demonstrates how GPTs may help accelerate learning from incidents through models of operational failure. The motivation for this research is discussed by Harel (2025).

Objective: Coping with the Unexpected

According to common practices, predictable accidents can be avoided by employing protective methods. A key industry challenge is to develop a methodology, comprising methods and procedures, to protect against the unexpected.

The objective of this research is to demonstrate a way to support common engineering practices, such as agile development, in operation design.

Large Language Models (LLMs) like GPT can significantly enhance incident analysis by automating, accelerating, and augmenting critical steps in understanding and learning from operational failures, especially in complex environments (Gao et al., 2023).

Methodology: Probing the Exceptions

Models of error generation are complex, and the costs of error prevention are extremely high. Real-time monitoring and feedback from past failure events facilitate the identification of design mistakes. Applying cross-domain machine learning enables gradual, agile, and cost-effective development of a generic model of error generation, ultimately achieving safe, seamless operation.

Rare events, such as catastrophic accidents, system failures, or near-misses, pose a paradox in system design and error prevention: they are high-impact but low-frequency, making them difficult to study, anticipate, or prevent through conventional learning methods (Perrow, 1984).

Incident analysis enables developing models by extracting patterns, causality, and system weaknesses from real or near-miss operational events. This approach turns past failures into proactive design improvements, helping to create models that predict and prevent similar errors in future system operations (Hollnagel, 2004).

Event tracking is the systematic capture of data about actions, system states, user interactions, and contextual factors leading up to, during, and after an incident. Effective event tracking is essential for reconstructing the sequence of events and identifying causes and system weaknesses (NIST SP 800-92). Event tracking is essential, as errors are often unexpected and sometimes unpredictable. A basic requirement is that activity before and during the incident has been traced and recorded.

Detecting exceptions: Probes are tools, mechanisms, or code snippets embedded into systems to monitor behavior, detect anomalies, and flag exceptions as they occur. They are especially critical in complex, safety-critical, or automated environments where silent failures or unexpected conditions can go unnoticed without active observation (Kiczales et al., 1997).

Rule development: The methodology of rule development is based on gradual rule definition and validation (D&V) using case studies. GPTs can assist with the analysis of case studies and with formalizing them in terms of operational rules.

Outcomes of this Research: Feasibility

The goal of this research is to demonstrate a way to implement the methodology outlined here. The cases studied in the research are the MX981 (DeHart, 2007) and Proton M (Siddiqi, 2015) incidents, and the PL603 accident (Letts, 1996).

- Case selection is demonstrated by the well-known Murphy's incident during the testing of the MX981 operation.
- The errors identified in the MX981 case concern the proper way to install particular sensors during system assembly, and ways to enable early detection of errors in sensor assembly by referencing limits that define normal measurements.
- The rules in the MX981 case are abstractions of the rules of sensor assembly, namely, choosing sensors with asymmetric sockets, and rules of error detection, namely, by expressing the limits in terms of parameters.
- Cross-industry expansion is demonstrated by the Proton M case study, which repeats the sensor assembly error involved in the MX981 case. (This expansion is demonstrated by Harel (2024a).)

- Cross-domain extension is demonstrated in the analysis of the PL603 accident. In this case, the rule about the range of sensory data is extended to apply to any risk indicator (Harel, 2024b).

Operation Design in Practice

Sampling: Select challenging incidents, such as celebrated accidents, which may represent a category of incidents.

AI support: According to the Black Swan theory, system design cannot protect against the unpredictable. AI tools may identify important incidents and the quality of activity tracking:

- Identify the operational error based on analysis of the reported investigations. Using design-oriented AI tools, we may obtain a set of hypothetical errors involved in the incident generation.
- Identify the design features that enable the symptoms, namely, the operational errors. Using software-oriented AI tools, we may identify problems in the software design and in the code, which might result in errors.
- By expressing the design problems in generic terms, cross-industry, deep-exploration AI tools may expand the conclusions obtained about the original incident and apply them to other industries, enabling interaction designers to avoid these errors elsewhere.

Conclusions

AI-driven approaches allow for early detection and prevention, reducing the operational and vendor costs associated with traditional error management.

System developers can use this methodology to design error-proof operations, detect unnoticed exceptions, and enforce normal operation definitions.

The cross-domain applicability of GPT-driven insights accelerates learning and the implementation of best practices, making safety and reliability improvements more accessible and affordable.

To speed up learning, system design should incorporate means for detecting and reporting unnoticed exceptions.

To facilitate the development of cost-effective error prevention, system design should also incorporate means for enforcing operation according to the definition of normal operation.

References

DeHart, R. L. (2007). Project MX-981: John Paul Stapp and deceleration research. *Aviation, Space, and Environmental Medicine*, 78(5), 516-520

Gao et al. (2023). LLMs for Root Cause Analysis in IT Logs, arXiv

Harel, A. (2024a), Assembly verification: the Proton M case study, DOI: 10.13140/RG.2.2.14397.88808

Harel, A. (2024b). Enforcing feature availability: the PL 603 case study, DOI: 10.13140/RG.2.2.34802.52165

Harel, A. (2025). Operations Engineering in the 5th Industrial Revolution, DOI: 10.13140/RG.2.2.32819.13603

Hollnagel, E. (2004). *Barriers and Accident Prevention*. Ashgate.

Kiczales, G. et al. (1997)**.** *Aspect-Oriented Programming* – injecting probes via instrumentation

Letts, Q. (1996), "Aeroperu B757 Accident," The Times, October 3,

NIST SP 800-92: Guide to Computer Security Log Management

Perrow, C. (1984). Normal Accidents: Living with High-Risk Technologies.

Siddiqi, A. A. (2015). Launch vehicle failures in the post-Soviet era: A review. *Acta Astronautica*, 109, 196–205. DOI: 10.1016/j.actaastro.2014.12.024