Integrating STPA with Large Language Models

An Engineering Examination Through a Smoke Alarm Case Study

Abstract

Large Language Models (LLMs) are increasingly being considered as engineering assistants for requirements analysis, knowledge retrieval, scenario generation, design exploration, and decision support. System-Theoretic Process Analysis (STPA) offers a structured approach to system safety based on control, feedback, system states, and safety constraints rather than on component failures alone. This paper examines three interpretations of STPA-LLM integration: LLMs as assistants to an STPA analyst, STPA as a language-modeling problem, and STPA applied to systems that contain an LLM. A residential smoke alarm is used as a case study. The analysis shows that hazards may emerge from interactions among alarm states, maintenance, nuisance alarms, and human response. The paper argues for a hybrid approach: LLMs can broaden the analyst's search space, but explicit system models and STPA remain the basis for causal reasoning, validation, and safety judgment.

Keywords: *STPA, Large Language Models, LLM, system safety, systems engineering, human-systems integration, sociotechnical systems, unsafe control actions, safety constraints, smoke alarms.*

1. Introduction

Large Language Models have introduced a new class of engineering tools capable of processing substantial textual information, generating structured responses, identifying patterns, and assisting knowledge-intensive work. Reported applications include requirements engineering, document analysis, design exploration, and decision support [7, 8]. This development raises a natural question for system safety engineering: can an LLM support, or potentially perform, System-Theoretic Process Analysis (STPA)?

STPA is not simply a procedure for producing a list of hazards. Derived from the System-Theoretic Accident Model and Processes (STAMP), it understands safety as constraints imposed on system behavior [1, 2]. Its focus is the control structure: controllers, controlled processes, control actions, feedback, process models, and system states. LLMs can produce plausible textual descriptions without necessarily representing these causal relationships. Moreover, factuality and unsupported generation remain known limitations of current LLMs [5, 6].

> ***What role can an LLM legitimately play within an engineering process whose validity depends on a structured representation of system control and causality?***

2. STPA and the Nature of System-Theoretic Analysis

Traditional safety analysis frequently emphasizes component failures and event propagation. STPA takes a broader systems perspective: unsafe behavior can emerge from interactions among technical, human, organizational, and operational elements, including when components operate as designed [1]. Controllers impose constraints on controlled processes, while feedback informs the controller about process state [1, 2].

STPA identifies losses, hazards, system-level safety constraints, a control structure, unsafe control actions (UCAs), and scenarios leading to those UCAs [2]. A control action is judged in context, not in isolation. The STPA Handbook identifies four UCA categories: a required action is not provided; an unsafe action is provided; an action is provided too early, too late, or in the wrong sequence; or an action lasts too long or stops too soon [2].

3. Three Interpretations of STPA-LLM Integration

3.1 LLM as an Assistant for Performing STPA

The conservative interpretation retains STPA as the engineering method and uses the LLM as an assistant. It can propose hazards, identify candidate system elements, formulate scenarios, suggest control actions, and search documentation. Such output remains candidate propositions, not a validated STPA analysis. Because LLM factuality remains an active research problem [5, 6], the defensible claim is that an LLM may support selected activities within STPA, not independently perform STPA.

3.2 STPA as a Language-Modeling Problem

A more ambitious approach would encode STPA knowledge in an LLM and infer results directly from textual system descriptions. This assumes that learned textual regularities preserve the causal control structure relevant to safety. That equivalence should not be assumed: similar terminology can describe different causal structures, and similar systems can require different constraints because operational contexts differ.

3.3 STPA Applied to Systems Containing LLMs

A third interpretation reverses the question. An LLM can be treated as a component of a sociotechnical control structure: it receives information, generates outputs, and influences human or machine decisions. Its outputs can therefore be control actions or inputs to other controllers. Evaluation should consider autonomy, impact, validation requirements, and human oversight [8]. The STPA question becomes: what UCAs can arise from interactions among an LLM, its users, its information sources, and the controlled system?

4. Case Study: A Smoke Alarm

4.1 System Description

A residential smoke alarm is simple enough to understand while still exhibiting important sociotechnical interactions. A simplified system contains a smoke sensor, power source, decision logic, alarm, maintenance function, and human user. The safety objective is to detect hazardous smoke conditions and provide an effective warning. U.S. Fire Administration and NFPA guidance emphasizes regular testing, maintenance, battery replacement, and avoiding disabling alarms [10, 11].

4.2 System Losses

- L1. Loss of timely warning of a fire.
- L2. Delayed warning of a fire.
- L3. Loss of confidence in the alarm system.
- L4. Loss of alarm availability due to inappropriate maintenance or disabling.

Empirical work reports disconnected or absent batteries among important causes of nonfunctional alarms; unwanted alarms and low-battery warnings can also contribute to disabling behavior [12, 13].

5. STPA Analysis of the Smoke Alarm

5.1 Hazards

- H1. A fire is present while the smoke alarm is unavailable to provide an effective warning.
- H2. The alarm does not provide an effective warning when a hazardous fire condition exists.
- H3. The alarm produces repeated unnecessary warnings that cause inappropriate user response.
- H4. The system indicates or implies operational availability when its safety function has not been verified.

These hazards are stated at system level. Battery failure, for example, is not itself the hazard; the safety-relevant condition is loss of warning while a fire exists [1, 2].

5.2 Unsafe Control Actions

UCA Category 1: Required action not provided

The system fails to generate an alarm when hazardous smoke is present. This may arise through power loss, disabling, an inappropriate state, or an incorrect process model - not only through a defective sensor.

UCA Category 2: Unsafe action provided

The alarm activates when conditions do not justify an emergency warning. Repeated nuisance alarms matter because they may drive inappropriate user response [12].

UCA Category 3: Correct action at the wrong time

The system returns to an apparently normal state before its ability to perform the safety function has been established. Following a transient interruption, normal indication before verification of detection capability is unsafe.

UCA Category 4: Incorrect duration

An alarm may be silenced too long, or a suppression function may remain active beyond its justified context. A hush function can be legitimate, but its duration is safety-relevant.

6. Critical Scenarios

6.1 Transient Power Loss

A short power interruption can temporarily remove detection or communication capability. Restoration of power does not necessarily restore the safety function. If the system transitions from “power restored” to “normal operation” without verifying the relevant safety function, a hazardous intermediate condition may remain hidden.

6.2 Improper Battery Replacement

A user may believe that replacing a battery restores normal service, although correct installation, electrical contact, sensor operation, and testing have not been confirmed. The relevant control chain is maintenance action -> system state -> verification -> return to service.

6.3 Repeated Nuisance Alarms

The sociotechnical sequence is nuisance alarm -> reduced trust -> altered user response -> alarm disabling or ignoring -> loss of safety function. The literature supports the relevance of this mechanism [12, 13].

7. Safety Constraints

- SC1. The system shall not indicate normal operational availability unless detection and alarm capability have been established for the current state.
- SC2. The system shall clearly distinguish normal operation, maintenance, degraded capability, and unavailable states.
- SC3. A temporary hush or suppression function shall be time-limited and shall not conceal a continuing hazardous condition.
- SC4. Return to service following power loss or maintenance shall be explicit and subject to appropriate verification.

These constraints shift the focus from “what failed?” to “what system behavior must not occur?” - the central move of STPA [1, 2].

8. What Can an LLM Contribute to the Analysis?

For the smoke-alarm case, an LLM can generate candidate hazards, suggest user interactions, enumerate maintenance states, propose UCAs, and formulate questions about power interruptions, nuisance alarms, and testing. This is especially valuable when an analyst seeks to widen the search space.

The risk is an illusion of completeness. A long, fluent list is not necessarily a complete analysis. The LLM can expand hypotheses; the engineering model must determine which hypotheses are possible, relevant, and constraint-violating in the actual system. Human review and traceability are therefore essential [7, 8].

9. Language Model versus System Model

A language model represents relationships among expressions, concepts, and linguistic patterns. A system model represents entities, states, control relationships, feedback, and interactions relevant to behavior. This does not mean that an LLM cannot help construct a system model; it may be valuable for eliciting and organizing candidate knowledge. It does mean that linguistic plausibility must not be treated as an automatic substitute for a causal system model.

A practical hybrid architecture is therefore to use the LLM as an exploration and hypothesis-generation layer, then test each candidate scenario against the structured system model: do the components exist, is the state reachable, is the proposed control chain present, and does the action violate a safety constraint?

10. Discussion

The smoke-alarm example is deliberately simple. It nevertheless demonstrates a central limitation of purely text-based safety analysis. An LLM can generate a convincing description, hazards, and scenarios, but the key question is whether the output represents the system's control mechanism and the conditions in which behavior becomes unsafe.

Nuisance alarms show why systems thinking matters. Treating each unwanted alarm merely as a detector fault can hide the broader chain: repeated alarms, diminished trust, alarm disabling, and eventually failure to warn of fire. As LLMs become more capable of producing professional-looking explanations, the distinction between linguistic quality and engineering validity becomes increasingly important.

11. A Hybrid STPA-LLM Workflow

- Step 1: Use the LLM to interrogate available documents and generate candidate losses, hazards, states, controllers, and scenarios.
- Step 2: Build or update an explicit control-structure model and process-model assumptions.

- Step 3: Conduct STPA on that model, recording UCAs, contexts, causal scenarios, and safety constraints.
- Step 4: Use the LLM to challenge omissions, explain model elements, or suggest testable scenarios; validate every candidate against the model and subject-matter expertise.
- Step 5: Maintain provenance and review records so that generated content is distinguishable from verified engineering conclusions.

This workflow assigns LLMs a bounded but useful role: they improve discovery and communication, while the structured model remains the source of safety judgment.

12. Validation and Human Oversight

Validation must be proportional to the consequence of use. For safety-relevant applications, generated assertions should be traceable to authoritative documentation, model elements, test evidence, or expert judgment. Evaluation should examine not only whether an LLM produces plausible UCAs, but also its omissions, false positives, sensitivity to prompt wording, and behavior when requirements are incomplete [5, 6, 8].

Human oversight is not a ceremonial sign-off. It includes defining the system boundary, resolving ambiguity, judging reachability and causal adequacy, approving safety constraints, and deciding whether evidence supports a conclusion. Those activities are central to STPA and cannot be established by fluent text alone.

13. Future Research

Future work should compare LLM-generated candidates with analyses produced by experienced STPA practitioners across a range of systems. Useful measures include recall of known hazards and UCAs, false-positive burden, traceability to model elements, quality of causal scenarios, and the time required for expert verification. Research should also examine representations that link LLM outputs to explicit control models, requirements repositories, simulations, and assurance cases [3, 4, 9].

The productive question is not simply whether an LLM can perform STPA. It is: in which tasks within an STPA workflow does an LLM deliver measurable value, and in which tasks are structured models or expert judgment indispensable?

14. Conclusions

STPA and LLMs should not be treated as competing methods. LLMs are useful for processing linguistic knowledge, generating possibilities, and broadening exploratory analysis. STPA provides the engineering framework for analyzing control, interactions, and safety constraints. The smoke-alarm case shows that even a simple system can contain hazards that are not understood fully through component failures alone.

The promising direction is hybrid: LLMs for knowledge generation and hypothesis formation, explicit system models for structural and causal representation, and STPA for framing the safety question and assessing control constraints. The challenge is not to declare an LLM a safety engineer, but to design an engineering process in which language capability strengthens - rather than substitutes for - validated system reasoning.

References

[1] N. G. Leveson, Engineering a Safer World: Systems Thinking Applied to Safety. MIT Press, 2011.

[2] N. G. Leveson and J. P. Thomas, STPA Handbook. MIT Partnership for a Systems Approach to Safety and Security, 2018.

[3] N. G. Leveson, "A New Accident Model for Engineering Safer Systems," Safety Science, vol. 42, no. 4, pp. 237-270, 2004.

[4] J. P. Thomas, Extending and Automating a Systems-Theoretic Hazard Analysis for Requirements Generation and Analysis. PhD dissertation, MIT, 2013.

[5] Z. Ji et al., "Survey of Hallucination in Natural Language Generation," ACM Computing Surveys, vol. 55, no. 12, 2023.

[6] L. Huang et al., "A Survey on Hallucination in Large Language Models: Principles, Taxonomy, Challenges, and Open Questions," ACM Transactions on Information Systems, 2025.

[7] M. J. H. M. M. et al., "Large Language Models for Software Engineering: Survey and Open Problems," arXiv:2310.03533, 2023.

[8] INCOSE, Systems Engineering Vision 2035. International Council on Systems Engineering, 2022.

[9] A. Abdulkhaleq and S. Wagner, "A Systematic Literature Review of Safety Requirements Methods," arXiv:1508.03536, 2015.

[10] U.S. Fire Administration, "Smoke Alarms." Federal Emergency Management Agency, accessed 2026.

[11] National Fire Protection Association, "Smoke Alarms: Safety Tips." NFPA, accessed 2026.

[12] R. F. Fahy and J. M. Molis, Smoke Alarms in U.S. Home Fires. National Fire Protection Association, 2014.

[13] M. Ahrens, Smoke Alarms in U.S. Home Fires. National Fire Protection Association, 2021.