From STPA to STPD: Requirements for Error-Tolerant Operational Design

Battery Replacement as a Case Study

Avi Harel, Ergolight

ergolight@gmail.com

Citation-integrated working manuscript | 2 November 2026 – pre print

Abstract

System-Theoretic Process Analysis (STPA) provides a structured approach to unsafe control actions, causal scenarios, and safety constraints, while system-theoretic research emphasizes the role of interactions and inadequate control in losses (Leveson, 2012; Leveson & Thomas, 2018). Yet operational design remains vulnerable to omissions and incomplete assumptions under changing contexts (Rasmussen, 1997; Woods, 2015). Building on STPD as introduced by Harel (2026), this article proposes twelve methodological requirements organized into design-time exception prevention and runtime operational enforcement. Design time specifies operational objectives, contexts, scenario constraints, transitions, and verification rules. At runtime, Canonical Scenario Representations (CSR) and Controller Multi-Service Interactions (CMSI) apply the models, while decision support, proactive troubleshooting, protective responses, and design-mistake investigation address changing and unexpected conditions. A hypothetical smoke-alarm restoration scenario and the historically reported Kandahar GPS battery-replacement event illustrate the distinction between equipment power/function and readiness for the intended task (Harel, 2026; Loeb, 2002; U.S. Fire Administration, n.d.). Established work on virtual sensing and model-based fault diagnosis provides supporting technical mechanisms for deriving otherwise unobserved operational variables (Doshmanziari et al., 2020; Mattera et al., 2018). The contribution is a proposed, traceable STPD design framework, not an experimentally demonstrated reduction in mistakes relative to rigorous STPA.

1. Introduction

1.1. Operational failures and design mistakes

Operational losses frequently involve human action, technical malfunction, environmental disturbance, and inadequate coordination. STAMP interprets accidents through insufficient control of system behavior rather than component failure alone (Leveson, 2012). This article adopts a preventive engineering principle: investigate the design provisions that failed to prevent, detect, contain, or recover from operational deviations. This attribution is a methodological stance, not empirical proof that every observed loss has a uniquely identifiable design mistake. Rasmussen (1997) likewise emphasized control, adaptive behavior, and boundaries of acceptable performance in dynamic sociotechnical systems.

1.2. Human fallibility and the anticipation dilemma

Designers may overlook tasks, modes, dependencies, and operating conditions; the resulting model may be internally consistent but incomplete. STPA offers systematic analysis but still depends on a valid control structure, selected losses and hazards, and analyst judgments about context and causal scenarios (Leveson & Thomas, 2018). The impossibility of predicting every combination of surprises motivates design provisions that remain effective as conditions vary. Woods (2015) characterizes resilience in part through graceful extensibility and sustained adaptability when surprise challenges operating boundaries. Murphy's Law is used here as a design heuristic rather than a quantitative empirical law.

1.3. STPA and the STPD extension

STPA explicitly analyzes control actions, feedback, process models, and losses; its scope may include performance or mission losses in addition to injury and damage (Leveson & Thomas, 2018). Therefore, STPD should not be presented as introducing non-safety losses to systems engineering for the first time. Harel (2026) proposed STPD as a model-based extension in which scenario-centered coordination, CSR, CMSI, and runtime checks connect constraints to operation. The present manuscript formalizes additional procedural obligations; these twelve requirements are proposals of this article, not established provisions of the STPA Handbook.

1.4. Design time and runtime

During design time, STPD proposes to specify intended services, operational goals, context, permissible configurations, transition rules, and verification provisions. During runtime, CSR supplies an active scenario reference while CMSI coordinates multiple services with that reference (Harel, 2026). The intended aim is prevention of incoherence before it becomes an unacceptable failure. An encountered context deserves protection whether predicted or not. In this article, an operational exception denotes an inability to establish or sustain required operational coherence, not merely unfamiliarity or novelty. This definition is a proposal of STPD, not a standard definition from STPA.

1.5. Coherence, seamless operation, and research objective

Operational coherence denotes compatibility of task, context, controller decisions, service configurations, and actual behavior. Seamless operation denotes effective task achievement with avoidable interruption and coordination burden minimized. Related Safety-II and resilience research examines how successful performance is sustained amid variability (Hollnagel, 2014; Woods, 2015), and classic automation research cautions against shifting abnormal-condition burdens to human operators (Bainbridge, 1983). The research question is whether explicit design-time rules and runtime model enforcement provide a systematic additional defense against design mistakes beyond conventional STPA. The following sections derive requirements and examine them through battery replacement.

2. From STPA to STPD

2.1. What STPA already provides

STPA derives unsafe control actions and loss scenarios from a control structure, and includes problems such as incorrect process models, inadequate feedback, timing, and coordination (Leveson & Thomas, 2018). It can be applied beyond traditional physical safety losses. Any methodological comparison must preserve those established capabilities and distinguish them from incomplete applications of the method.

Figure 1 situates STPD as the operational-design extension of the STAMP/STPA sequence. STAMP frames safety as a control problem; STPA identifies losses, hazards, unsafe control actions and safety constraints; STPD proposes to translate relevant constraints into coordinated, enforceable operational provisions. This is a methodological emphasis rather than a claim that STPA cannot inform design implementation. Effective constraint implementation requires appropriate enforcement and evidence that the constraint is satisfied.

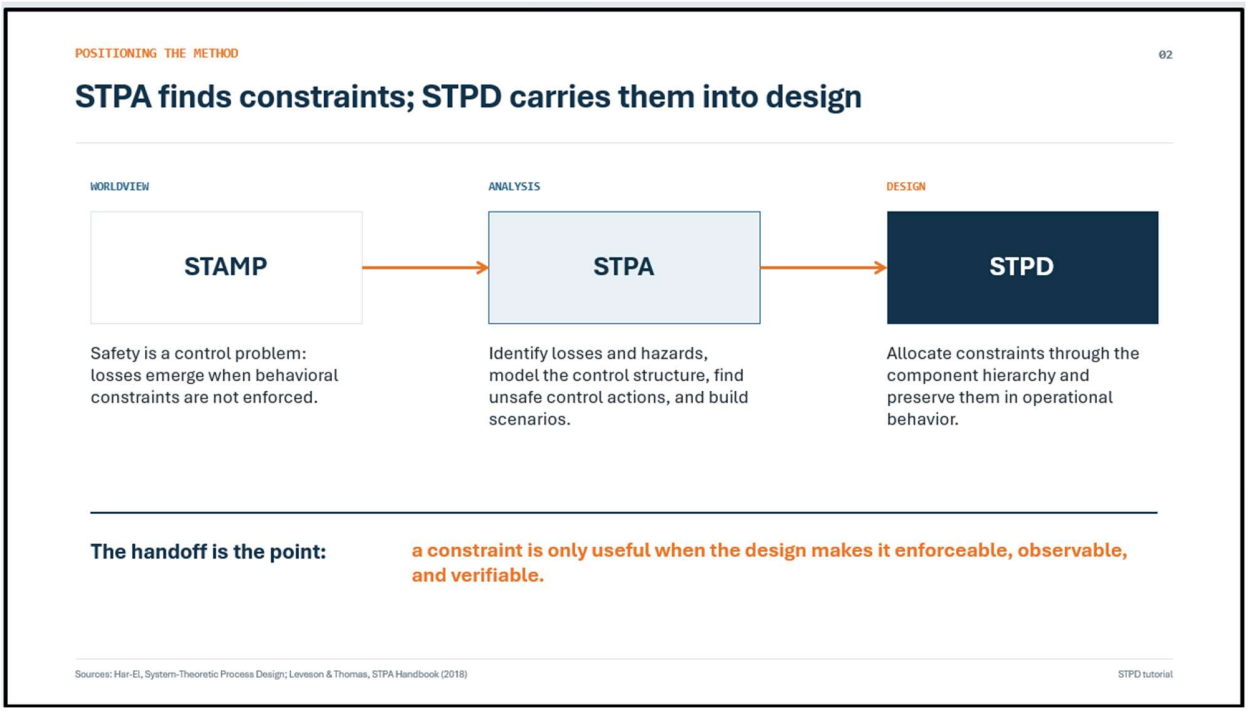


Figure 1. From STAMP through STPA to STPD: translating safety constraints into operational design. Adapted from the author-supplied diagram; emphasis on enforcement, observability, and verification.

2.2. Design-time model-based prevention

The proposed STPD design-time contribution is to derive required configurations from intended tasks, then define permissible transitions and completion criteria without first listing every dangerous initiating event. Such use of operational models is consistent with the broader system-theoretic emphasis on constraints and boundaries (Rasmussen, 1997; Leveson, 2012). It remains a hypothesis that making these obligations explicit reduces design mistakes in practice.

Figure 2 presents the proposed development cycle for operational coherence. At design time, STPD organizes function-oriented operational design and model-based coherence with complementary hazard-analysis and exception-control methods. Deployment connects the design framework to runtime operation. Accidents, near misses, and detected exceptions provide feedback for design investigation and correction. The figure illustrates a proposed engineering workflow, not empirical evidence that all deviations can be prevented.

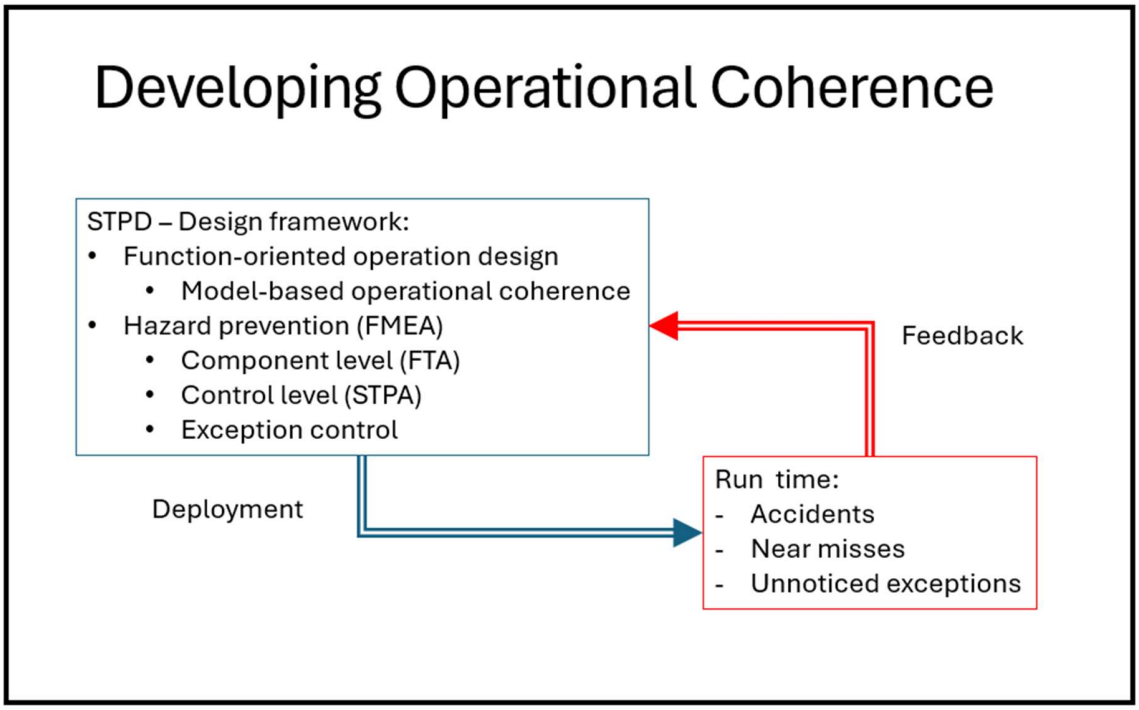


Figure 2. Developing operational coherence: design framework, deployment, runtime outcomes, and feedback (author-supplied conceptual figure).

Interpretation note. FMEA, FTA, and STPA are complementary analysis approaches; the diagram's hierarchy is conceptual rather than a strict classification of their technical scope. An unnoticed exception cannot directly produce feedback until it is identified through monitoring, subsequent evidence, or investigation. The observability obligation in R11 addresses this gap.

Figure 3 elaborates the design-time architecture by showing how generic CSR and CMSI models and a rule infrastructure are customized during functional design. Operational coherence is pursued through a model of system operation, scenario-based complexity control, rule-based constraints, and decision support. Hazard analysis, including FMEA, component reliability and redundancy analysis, and STPA identification of unsafe control actions, contributes exception-control requirements and feeds findings back into functional design. The figure therefore expresses integration of safety activities within functional design, rather than a one-way safety review after design completion.

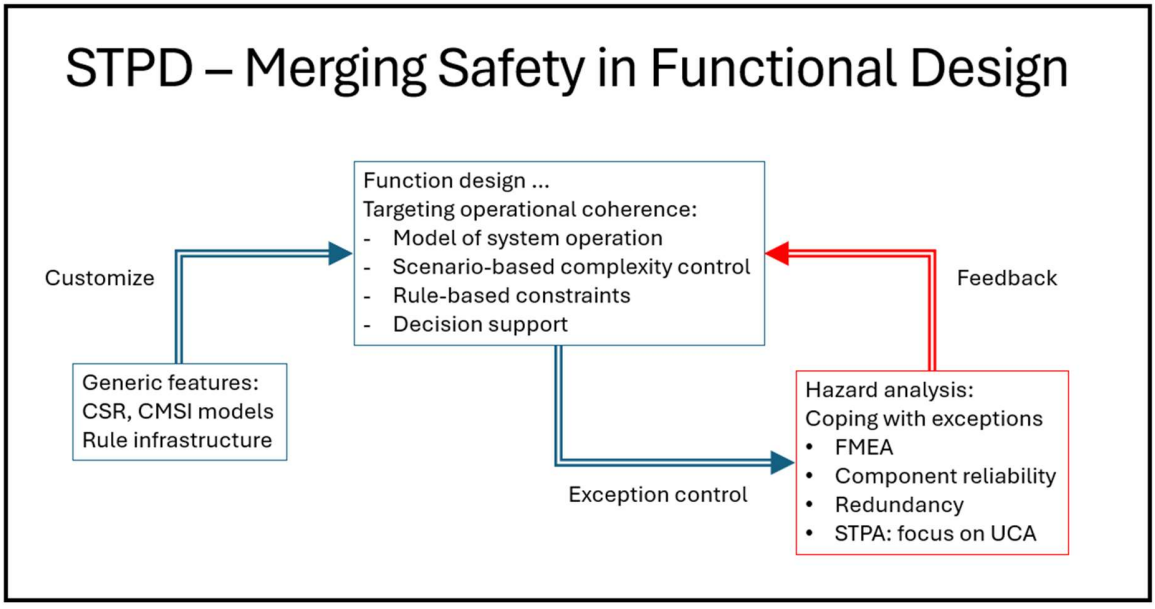


Figure 3. STPD: Merging safety in functional design. Generic CSR/CMSI features are customized for model-based operation; hazard analysis supplies exception-control requirements and feedback (author-supplied conceptual figure).

Interpretation note. The right-hand block combines analysis methods and engineering measures. FMEA and STPA are analysis methods, while component reliability and redundancy are design or assurance concerns. The feedback arrow represents the iterative revision of functional design based on hazard and exception analysis.

2.3. Runtime CSR and CMSI

Harel (2026) presents CSR as a scenario reference and CMSI as a coordination structure for controller interactions with multiple services. In the present proposal, runtime is required to use those models under encountered contexts, rather than treating unanticipated work as exempt from protection. If coherence cannot be established, an exception response is required. The architecture should distinguish uncertain context identification from demonstrated configuration mismatch; both may demand protective action.

2.4. Decision support, observability, and learning

Runtime decision support assists controllers in evaluating permissible actions, resources, and consequences. Proactive troubleshooting calls for early indications of degraded operation, aided where useful by inferred variables or virtual sensors. The concept builds on established fault-detection research and is not itself novel to STPD (Mattera et al., 2018; Doshmanziari et al., 2020). Woods (2015) and Hollnagel (2014) support the broader value of sustaining performance under changing conditions. The STPD-specific proposal is to connect these capabilities to an explicit scenario model and to investigate unexpected exceptions as possible evidence of design deficiencies.

3. Battery Replacement as an Operational Case Study

3.1. Generic restoration problem

Maintenance can alter power, retained configuration, mode, interfaces, and the assumptions of other system participants. The relevant design distinction is between restoring physical power, verifying a service capability, and establishing task-compatible operational readiness. The latter two cannot automatically be inferred from the first. The source smoke-alarm manuscript explicitly includes improper battery replacement and return-to-service verification as analyzable STPA scenarios (the supplied smoke-alarm working study).

3.2. Residential smoke alarm: hypothetical case

For a residential alarm, normal monitoring requires the availability of detection and occupant warning capabilities. The U.S. Fire Administration (n.d.) stresses that alarms require correct installation, maintenance and regular testing, and that alarms with missing or dead batteries are ineffective. In the hypothetical scenario here, battery insertion restores power while complete readiness remains unverified. The proposed design shall not equate battery insertion with established capability. It should specify what available tests establish and where uncertainty remains. This is a hypothetical design demonstration, not an empirical account of a particular alarm model.

3.3. Kandahar GPS: documented historical mechanism

Contemporary reporting on the December 2001 Kandahar friendly-fire event attributed the targeting-coordinate error to a GPS receiver that, after battery replacement, displayed its own location; its operator reportedly did not recognize this reset behavior (Loeb, 2002). A RAND historical study independently recounts the battery-change and own-position reset mechanism (Lambeth, 2005). The mechanism supports examining mode and context compatibility, but this manuscript does not claim that an STPD implementation existed or was experimentally tested during that event.

3.4. Common implication

In both cases the central question is whether a service is compatible with its current task after a configuration-changing intervention. The smoke-alarm case concerns verified functional readiness; the GPS case concerns the meaning of output relative to task context. Both can be analyzed with STPA, and both provide examples from which the proposed STPD requirements may be derived (Leveson & Thomas, 2018; Harel, 2026).

4. STPD Methodological Requirements

4.1. Derivation and status

The twelve statements below are normative requirements proposed in this article. Harel (2026) motivates the CSR/CMSI architecture; Leveson and Thomas (2018) provide the STPA baseline; resilience and model-based diagnosis literature supports selected mechanisms (Hollnagel, 2014;

Woods, 2015; Mattera et al., 2018). These sources do not establish the completeness, exclusivity, or empirical efficacy of the proposed twelve-requirement set.

4.2. Design-time prevention requirements

R1. Operational goal definition. STPD shall require: Specify tasks, objectives, acceptance criteria and priority of safety, performance, availability and UX.

R2. Operational coverage. STPD shall require: Inventory services and interventions including maintenance, testing, interruption and degradation.

R3. Context completeness. STPD shall require: Represent task-related external conditions and required internal modes, resources and dependencies.

R4. Scenario configuration and coordination rules. STPD shall require: Define configurations and applicable constraints that establish coherent operation.

R5. Transition integrity. STPD shall require: Define entry, intermediate, completion and exception conditions for scenario changes.

R6. Restoration integrity. STPD shall require: Require adequate evidence of destination-scenario readiness before service restoration is declared.

R7. Operational model verification. STPD shall require: Check coverage, traceability, consistency, unresolved assumptions and external validation.

R1–R7 derive from the proposed requirement that an operational model state what successful task performance requires. Comparable constraints may be derived by competent STPA analysts; the proposed STPD contribution is the mandatory, traceable integration of these design artifacts (Leveson & Thomas, 2018; Harel, 2026).

4.3. Runtime requirements

R8. CSR-based scenario coordination. STPD shall require: Maintain or re-establish the active context reference for encountered tasks, including unexpected situations.

R9. CMSI coordination and enforcement. STPD shall require: Coordinate multiple services and constrain actions to compatible scenario configurations.

R10. Model-based decision support. STPD shall require: Support controllers with available alternatives, constraints, predicted implications and uncertainty.

R11. Proactive troubleshooting. STPD shall require: Provide observability, inferred variables, emerging-deviation detection, diagnosis and early intervention.

R12. Exception protection and design-mistake investigation. STPD shall require: Protect when coherence cannot be established and investigate unexpected exceptions, verifying corrective actions.

The CSR/CMSI concepts derive from the STPD INSIGHT article (Harel, 2026). Virtual sensing and diagnostic inference are independently established technologies (Mattera et al., 2018;

Doshmanziari et al., 2020). R12 combines immediate runtime response with slower design feedback, activities that should be separately verified.

4.4. Traceability matrix

ID	Expected artifact	Verification question
R1–R3	Goal, intervention and context models	Are significant tasks, dependencies and goals grounded in external evidence?
R4–R6	CSR rules and transition/restoration tests	Are compatible configurations and return-to-service criteria executable and testable?
R7	Coverage/validation record	Have omissions, contradictory assumptions and model limitations been examined?
R8–R9	Active CSR and CMSI interface/control specification	Can actual situations be mapped to protected, coordinated operation?
R10	Decision-support specification	Are options and uncertainty constrained by operational goals?
R11	Observability and diagnostic specification	Can relevant emerging deviations be estimated or measured reliably?
R12	Protection and learning record	Are incoherence responses and corrective feedback tested and closed?

5. Comparative Assessment of STPA and STPD

5.1. Comparative logic

The comparison does not contrast a deliberately weak STPA analysis with an idealized STPD design. In the STPA Handbook, the analyst can recognize incorrect process models, maintenance effects, delayed feedback and inappropriate controls (Leveson & Thomas, 2018). The hypothesis is narrower: explicit STPD design artifacts may make omissions easier to detect and may improve the translation of analysis into runtime enforcement.

5.2. Design-time derivation of restoration requirements

First, the task specifies required capabilities. Second, those capabilities imply service configurations. Third, interventions such as battery replacement are mapped to affected configurations. Fourth, the return transition specifies completion conditions. Fifth, evidence is required before readiness is confirmed. This chain follows R1–R7 and can produce a restoration requirement without separately enumerating every initiating hazard. It is an analytical derivation, not a proof that STPA cannot produce the same requirements (Harel, 2026; Leveson & Thomas, 2018).

5.3. Runtime coordination under changed contexts

R8 and R9 require the active CSR and the CMSI implementation to constrain participating services according to task context. In the GPS case, the coordinate source would need to prove its interpretation compatible with targeting before use. In the alarm case, readiness would need to be established rather than inferred from power alone. Both are proposed design controls, not claims about devices in the historical record (Loeb, 2002; U.S. Fire Administration, n.d.).

5.4. Decision support and proactive troubleshooting

Decision support is proposed to expose alternatives and constraints to controllers. Proactive troubleshooting requires adequate observability and correction of emerging deviations. For a rigid fixed-volume gas container, an ideal-gas approximation estimates inventory $n = PV/(RT)$; a significant unexplained change may indicate leakage or another unmodeled transfer. This is a conceptual virtual-sensor example and depends on volume constancy, thermodynamic applicability, measurement uncertainty and legitimate flows. Peer-reviewed work demonstrates that model-based inference and virtual sensors can support fault detection, but not that this particular simplified estimator has been validated for all gas containers (Doshmanziari et al., 2020; Mattera et al., 2018).

5.5. Protection and design learning

If coherence cannot be established, R12 requires a response proportionate to task consequences and a subsequent investigation of possible design mistakes. The claim that any unexpected exception demands design review is a prescriptive STPD principle. It does not mean the event alone proves that a specific design mistake occurred. Work on resilience and sociotechnical control helps motivate careful management of surprises and variation (Rasmussen, 1997; Woods, 2015).

5.6. Comparative conclusion

The examples support the feasibility of deriving restoration and coordination constraints from intended operation. They do not measure omission frequency, analyst effort, or system outcomes. Thus, fewer design mistakes under STPD remains an experimentally testable hypothesis, not a demonstrated fact. A rigorous comparison should assess independently validated requirements and equivalent teams applying STPA and STPD.

6. Discussion

6.1. The anticipation dilemma and limits of models

The value of model-based enforcement is that a rule may prohibit a class of incompatible configurations without specifying every cause of incompatibility. The limitation is that the modeled rule can itself be incomplete. Rasmussen (1997) and Woods (2015) support the importance of adapting to moving boundaries and surprises. STPD must therefore verify model adequacy and provide uncertainty-sensitive responses instead of claiming that coherent representation guarantees correctness.

6.2. Seamless operation and human-system integration

Seamless operation requires more than excluding hazards: it requires reliable service, understandable feedback, effective decisions and manageable workload. Bainbridge (1983) showed that automation can transfer difficult abnormal-condition tasks to operators; Hollnagel (2014) argues for attention to how successful operation is sustained. STPD proposes to integrate these concerns with task-based CSR/CMSI coordination. This integration is the proposed contribution, not novelty of each individual technique.

6.3. Observability as a design obligation

Operational constraints are useful only when the system can establish whether the conditions hold. Virtual sensors and analytical redundancy can extend observability, as demonstrated in published fault-detection methods (Mattera et al., 2018; Doshmanziari et al., 2020). Still, observability and diagnosability are conditional on model and measurement quality. An unobservable requirement should trigger instrumentation or protective analysis, not an assumption that the condition is satisfied.

6.4. Verification and limits of evidence

The published STPD article introduces the conceptual architecture (Harel, 2026); the present twelve-requirement structure and comparative conclusions are new proposals. The historical Kandahar evidence establishes the reported battery-related GPS reset, not the efficacy of a hypothetical CSR intervention (Loeb, 2002; Lambeth, 2005). The smoke-alarm example is a design thought experiment. Future work should examine design completeness, safety/performance constraints, traceability, runtime diagnostics, human workload, and rates of missed requirements under independently specified test cases.

7. Conclusions

7.1. Main contribution

The article proposes twelve methodological requirements linking design-time models with runtime CSR/CMSI enforcement, decision support, proactive troubleshooting, exception protection, and design correction. This builds on the STPD architecture of Harel (2026) and the system-theoretic STPA foundation of Leveson and Thomas (2018). The primary claim is an alternative, structured derivation of operational constraints from intended task performance, not an exclusive inability of STPA to find related hazards.

7.2. What the case study supports

The smoke-alarm and Kandahar examples show why restored power or local functionality cannot automatically be equated with operational readiness. The historical GPS account is supported by contemporary reporting and a later RAND study (Loeb, 2002; Lambeth, 2005); smoke-alarm maintenance importance is supported by the U.S. Fire Administration (n.d.). These sources establish case relevance. They do not empirically demonstrate that STPD prevents design mistakes more effectively than STPA.

7.3. Future validation

A controlled study should compare experienced analysts following conventional STPA and analysts applying the proposed STPD obligations. Outcomes should include omitted requirements, valid protective constraints, verification evidence, scenario-transition behavior, model sensitivity, and operator decision burden. Error-free and seamless operation remain engineering goals; the effectiveness and costs of the proposed two-phase method require independent validation.

References

Bainbridge, L. (1983). Ironies of automation. Automatica, 19(6), 775–779. https://doi.org/10.1016/0005-1098(83)90046-8

Doshmanziari, R., Khaloozadeh, H., & Nikoofard, A. (2020). Gas pipeline leakage detection based on sensor fusion under model-based fault detection framework. Journal of Petroleum Science and Engineering, 184, 106581. https://doi.org/10.1016/j.petrol.2019.106581

Harel, A. (2026). STPD: Operational design as a model-based extension of STPA. INSIGHT, 29(3), 48–57. https://doi.org/10.1002/inst.70067

Hollnagel, E. (2014). Safety-I and Safety-II: The past and future of safety management. Ashgate.

Lambeth, B. S. (2005). Air power against terror: America's conduct of Operation Enduring Freedom. RAND Corporation. https://www.rand.org/pubs/monographs/MG166-1.html

Leveson, N. G. (2012). Engineering a safer world: Systems thinking applied to safety. MIT Press. https://mitpress.mit.edu/9780262016629/engineering-a-safer-world/

Leveson, N. G., & Thomas, J. P. (2018). STPA handbook. https://www.flighttestsafety.org/images/STPA_Handbook.pdf

Loeb, V. (2002, March 24). Friendly fire deaths traced to dead battery. The Washington Post. https://www.washingtonpost.com/archive/politics/2002/03/24/friendly-fire-deaths-traced-to-dead-battery/7ba7f3f3-a219-4e15-9936-0473f03c48e8/

Mattera, C. G., Quevedo, J., Escobet, T., Shaker, H. R., & Jradi, M. (2018). A method for fault detection and diagnostics in ventilation units using virtual sensors. Sensors, 18(11), 3931. https://doi.org/10.3390/s18113931

Rasmussen, J. (1997). Risk management in a dynamic society: A modelling problem. Safety Science, 27(2–3), 183–213. https://doi.org/10.1016/S0925-7535(97)00052-0

U.S. Fire Administration. (n.d.). Smoke alarms. Federal Emergency Management Agency. https://www.usfa.fema.gov/prevention/home-fires/prepare-for-fire/smoke-alarms/

Woods, D. D. (2015). Four concepts for resilience and the implications for the future of resilience engineering. Reliability Engineering & System Safety, 141, 5–9. https://doi.org/10.1016/j.ress.2015.03.018