SAFETY-CRITICAL SYSTEMS RECOMMENDED PRACTICE

J3187™ MAY2023

Issued 2022-02
Revised 2023-05

Superseding J3187 FEB2022

(R) System Theoretic Process Analysis (STPA) Recommended Practices for Evaluations of Safety-Critical Systems in Any Industry

RATIONALE

This document provides recommended practices regarding how System Theoretic Process Analysis (STPA) may be applied to safety-critical systems in any industry.

NOTE: This document is a replacement for the original release of SAE J3187 and has been modified to make it clear that it is applicable for all industries dealing with safety critical systems. This new version is streamlined in nature and is not automotive specific. The original Human-Machine Interactions (HMIs), Model-Based Systems Engineering (MBSE), and Safety of the Intended Functionality (SOTIF) sections have been removed and will be offered as separate independent appendices under the original SAE J3187 number, utilizing different suffix labeling.

TABLE OF CONTENTS

TO PLACE A DOCUMENT ORDER: **Tel: 877-606-7323 (inside USA and Canada)**
Tel: +1 724-776-4970 (outside USA)
Fax: 724-776-0790
Email: CustomerService@sae.org
SAE WEB ADDRESS: **http://www.sae.org**

For more information on this standard, visit
https://www.sae.org/standards/content/J3187_202305/

1. SCOPE

This document provides recommended practices regarding how System Theoretic Process Analysis (STPA) may be applied to safety-critical systems in any industry.

1.1 Purpose

This document provides information and lessons learned from experienced STPA practitioners about the use of STPA for the evaluation of control-based safety-critical systems. Appendices explore specific areas of STPA application, such as Human-Machine Interaction (HMI) analysis, medical devices, Model-Based Systems Engineering (MBSE), cybersecurity, automotive, etc., and provide examples for specific industries and/or topics. These appendices are published as separate documents.

2. REFERENCES

2.1 Applicable Documents

The following publications form a part of this specification to the extent specified herein. Unless otherwise indicated, the latest issue of SAE publications shall apply.

2.1.1 SAE Publications

Available from SAE International, 400 Commonwealth Drive, Warrendale, PA 15096-0001, Tel: 877-606-7323 (inside USA and Canada) or +1 724-776-4970 (outside USA), www.sae.org.

SAE J3016 Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles

Khastgir, S., Birrell, S., Dhadyalla, G., and Jennings, P., "The Science of Testing: An Automotive Perspective," SAE Technical Paper 2018-01-1070, 2018, https://doi.org/10.4271/2018-01-1070.

Okada, M. and Christensen, A. (2017, October 25). SAE Recommended Task Force - Group 4 Example.

Suo, D., Yako, S., Boesch, M., and Post, K., "Integrating STPA into ISO 26262 Process for Requirement Development," SAE Technical Paper 2017-01-0058, 2017, https://doi.org/10.4271/2017-01-0058.

2.1.2 Other Publications

Abrecht, B. (2016). A New Approach to Hazard Analysis for Naval Systems. MIT STAMP Workshop. Cambridge MA: MIT.

Crash Avoidance Metrics Partnership (CAMP) Automated Vehicle Research (AVR). (2016). Automated Vehicle Research for Enhanced Safety. Washington DC: U.S. Department of Transportation.

Fleming, C., Placke, M., and Leveson, N. (2013). Technical report: STPA analysis of NextGen interval management components: Ground interval management (GIM) and flight decn interval management (FIM).

France, M.E. (2017). Engineering for Humans - Human-Automation Interaction in STPA. MIT STAMP/STPA Workshop (p. 18/50). Cambridge: MIT.

France, M.E. (2017). Engineering for Humans: A New Extension to STPA. Cambridge MA: MIT - Master's Thesis.

Georgiou, O.B. (2017). Haptic In-Vehicle Gesture Controls. Proceedings of the 9th International Conference on Automotive User Interfaces and Interactive Vehicular Applications Adjunct (pp. 233-238).

Immersion Corporation. (2017). MAGNA and IPG Media Labs Haptic Ad Study. Immersion Corporation; https://www.immersion.com/study-unveils-touchsense-ads-lead-to-brand-lift/.

Kalra, N. and Paddock, S.M. (2016). Driving to safety: How many miles of driving would it take to demonstrate autonomous vehicle reliability? Transportation Research Part A: Policy and Practice.

Khastgir, S., Brewerton, S., Thomas, J., and Jennings, P. (2021). Systems Approach to Creating Test Scenarios for Automated Driving Systems. Reliability Engineering and System Safety.

Leveson, N. (2015). A systems approach to risk management through leading safety indicators. Reliability engineering and system safety.

Leveson, N. and Thomas, J. (2018). STPA Handbook. Cambridge MA: MIT.

Malloy, N.H. (2017). Integrating STAMP-Based Hazard Analysis with MIL-STD-882E Functional Hazard Analysis. MIT STAMP Workshop. Cambridge MA: MIT.

Martinez, P.D. (2017). Agency in Mid-air Interfaces. Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (pp. 2426-2439).

Martinez, R.S. (2015). A comparison of STPA and automotive FMECA. MIT STAMP Workshop. Cambridge MA: MIT.

Object Management Group. (2020). RAAML Examples. Risk Analysis and Assessment Modeling Language (RAAML)Examples (Informative). Milford, MA, USA: Object Management Group.

Object Management Group. (2021). RAAML. Risk Analysis and Assessment Modeling Language (RAAML) Libraries and Profiles. Milford, MA, USA: OMG.

OMG, O.M. (2021). Risk Analysis and Assessment Modeling Language (RAAML) Version 1.0. Milford, MA 01757 USA: OMG.

Pawlicki, T. (2018). Experiences with STPA in Radiation Therapy. MIT STAMP Workshop. Cambridge MA: MIT.

Pope, G. (2019). Risk Management Using Systemic Theoretic Process Analysis (STPA). Livermore, CA (United States): Lawrence Livermore National Lab (LLNL).

Reason, J. (1995). Understanding adverse events: human factors. Quality in Health Care, pp. 4:80-89.

Ressler, G.E. and Vernacchia, M.A. (2018). Designing in Safety Tutorial. International System Safety Society Conference Proceedings. Phoenix: ISSS.

Ressler, G.E. and Vernacchia, M.A. (2019). Helping Beginners Use System Safety Methodologies. International System Safety Conference ISSC37 (pp. 4, 131). Norfolk VA: ISSS - International System Safety Society.

Ribeiro, D.D. (2016). A Systems Approach to the Development of an Aircraft Smoke Control System. MIT STAMP. Cambridge MA: MIT.

Silvis-Cividjian, N., Verbakel, W., and Admiraal, M. (2018). Application of STPA in Radiation Therapy: a Preliminary Study. MIT STAMP Workshop. Cambridge MA: MIT.

Skedung, L.A. (2012). Feeling Small: Exploring the Tactile Perception Limits. Scientific Reports, 1-6.

Thomas, J. (2020). 8.4.1.7 When STPA Results Surprise You - An industry case study employing STPA, Fault Trees, FMEA, and HAZOP. MIT STAMP Virtual Workshop. Cambridge MA: MIT.

Thomas, J. and Gibson, M. (2020). Industry Trials to Evaluate STPA's Effectiveness and Practicality for Digital Control Systems. MIT STAMP Virtual Workshop. Cambridge MA: MIT.

Vernacchia, M.A. (2016). Defining Safety Requirements for Human-Machine Interactions. International System Safety Conference (p. 6). Orlando: ISSS.

Vernacchia, M.A. (2016). Interal General Motors System Safety Training. Milford MI: General Motors Company.

Vernacchia, M.A. (2017). System Concept Analysis Methods. Milford: GM Internal Document.

Vernacchia, M.A. (2018). Delivering Safety Through Early Analysis Methods. ISSC 36 Conference Proceedings (p. 13). Phoenix: ISSS.

Vernacchia, M.A. (2018). Integration of STPA into GM System Safety Process. MIT STAMP Workshop (p. 5). Cambridge MA: MIT.

Vernacchia, M.A. (2019). Integrating STPA into Large Organizations - Lessons Learned at General Motors. ISSC37 - International System Safety Conference. Norfolk VA: International System Safety Society.

Vincoli, J.W. (2005). Basic Guide to System Safety, Second Edition. John Wiley and Sons, Inc., Hoboken, NJ, USA.

Young, W. (2018). STPA in Cybersecurity. MIT STAMP/STPA Workshop. Cambridge: MIT.

3. DEFINITIONS

ACCIDENT: Unexpected and unplanned event or circumstance resulting in a loss. (Leveson, MIT)

ARCHITECTURE: Representation of the structure of the item or functions or systems or elements that allows identification of building blocks, their boundaries, and interfaces, and includes the allocation of functions to hardware and software elements. (ISO 26262)

CONCEPT OF OPERATIONS: A document produced early in the requirements definition process to describe what the system will do (not how it will do it) and why (rationale).

NOTE 1: Usually referred to as “ConOps.”

NOTE 2: It should also define any critical, top-level performance requirements or objectives (stated either qualitatively or quantitatively) and system rationale. (INCOSE Systems Engineering Handbook)

CONDITION: State of something, especially with regard to its appearance, quality, or working order. (ISSS working group)

CONTROL ACTION: A command or feedback item created by, or used by, a system element to perform its function(s).

NOTE: Functions “do things,” whereas, control actions are the commands/activities driven by the functions, and/or are the feedback needed by the element(s) to perform the function(s).

CONTROL STRUCTURE: Hierarchical structures where each level imposes constraints on the activities of the level beneath them and accidents are viewed as the consequence of inadequate control of safety constraints.

NOTE: (https://systemsthinkinglab.com/stamp/); provides a method to document and graphically depiction of the functional design of the system (Leveson); systems represented as interrelated components kept in a state of dynamic equilibrium by feedback loops of information and control. (Leveson)

DEDUCTIVE ANALYSIS: Analysis approach starting out with a general statement, or hypothesis, and examines the possibilities to reach a specific, logical conclusion; the general to the specific. (https://www.livescience.com/21569-deduction-vs-induction.html)

ELEMENT: System or part of a system including components, hardware, software, hardware parts, and software units. (ISO 26262)

EXPLORATORY ANALYSIS: Analysis approach that considers multiple points of view and tries to anticipate all possible objections to, or flaws in, a particular position, with the goal of seeking the truth; used when cause and effects may not be known or not known in sufficient detail. (https://en.wikipedia.org/wiki/Exploratory_thought)

FAIL SAFE: Characteristic of a system whereby any malfunction affecting the system safety will cause the system to revert to a state that is known to be within acceptable risk parameters. (FAA System Safety Handbook)

FAILURE: Termination of the ability of an element to perform a function as required. (ISO 26262)

FAULT: Abnormal condition that can cause an element or an item to fail. (ISO 26262)

HARM: Physical injury or damage to the health of persons. (ISO 26262)

HAZARD: Any set of system states, events, or conditions together with a particular set of environmental conditions that will lead to a loss.

HAZARD ASSESSMENT: Process involved in identifying and determining the hazard level. (Leveson)

HUMAN FACTORS (ERGONOMICS): Applied science concerned with designing and arranging interfaces people use so that both people and interfaces interact efficiently and safely.

INDUCTIVE ANALYSIS: Analysis approach using Inductive reasoning makes broad generalizations from specific observations, the specific to the general. (https://www.livescience.com/21569-deduction-vs-induction.html)

INTERACTION: System state affects or influences one or more vehicle systems, or system elements, even if they may not be connected or communicating with each other.

INTERFACE: Occurrence of direct data/info transfer and/or communication between two systems or system elements within a system.

LOSS: Something no longer possessed or having less of something. May include harm to people; damage or destruction to property or equipment; environmental harm, loss of mission objectives; etc. caused by losing something.

MALFUNCTION: A failure to function in a normal or satisfactory manner. (Oxford Languages)

MISHAP: An accident

REQUIREMENTS: Statements describing essential, necessary, or desired attributes. (FAA System Safety Handbook)

RISK: The combination of the probability of occurrence of harm and the severity of that harm. (ISO 26262)

RISK—ACCEPTABLE: Risk that is understood and agreed to by the program/project, governing authority, mission directorate, and other customer(s) such that no further specific mitigating action is required. (NASA Systems Engineering Handbook)

RISK—RESIDUAL: Risk remaining after the deployment of safety measures. (ISO 26262)

RISK—UNREASONABLE: Risk judged to be unacceptable in a certain context according to valid societal moral concepts. (ISO 26262)

RISK ASSESSMENT: Process of determining the risk level (quantifying risk). (Leveson)

RISK LEVEL: Function of the hazard level combined with (1) the likelihood of the hazard leading to an accident, and (2) hazard exposure or duration. (Leveson)

SAFETY: Absence of losses

SEVERITY: The magnitude or degree of consequences of an accident (mishap/adverse event)

SUB-SYSTEM: An element of a system that, in itself, may constitute a system. (FAA System Safety Handbook)

SYSTEM: Collection of elements that work together to achieve a common goal .

SYSTEM SAFETY ENGINEERING: Subdiscipline within Systems Engineering that deals with safety to identify safety-related risks and eliminate or control them.

SYSTEMS ENGINEERING: Engineering discipline whose responsibility is creating and executing an interdisciplinary process to ensure that the customer and stakeholder's needs are satisfied in a high quality, trustworthy, cost-efficient, and schedule compliant manner throughout a system's entire life cycle. (INCOSE)

TRADE OFF: Decision making actions that select from various requirements and alternative solutions on the basis of net benefits to the stakeholders. (INCOSE)

TRADE STUDY: A means of evaluating system designs by devising alternative means to meet functional requirements, evaluating these alternatives in terms of the measures of effectiveness and system cost, ranking the alternatives according to appropriate selection criteria, dropping less promising alternatives, and proceeding to the next level of resolution, if needed. (NASA Systems Engineering Handbook)

UNSAFE CONTROL ACTION: Control action that, in a particular context and worst-case environment, will lead to a hazard.

VERIFICATION: Confirmation that work products properly reflect the requirements specified for them ("you built it right"). (NASA Systems Engineering Handbook)

VALIDATION: Confirmation that the product fulfills its intended use in all of the environments that the product will be used in ("you built the right thing"). (NASA Systems Engineering Handbook)

4. ACRONYMS

CAST	Causal analysis based on system theory
DFMEA	Design failure mode and effects analysis
FIA	Functional interface analysis
FMEA	Failure mode and effects analysis
FTA	Fault tree analysis
HARA	Hazard analysis and risk assessment
HMI	Human-Machine Interaction
INCOSE	International Council on Systems Engineering
ISO	International Organization for Standardization
JASPAR	Japan Automotive Software Platform and Architecture
MBSE	Model-Based Systems Engineering
NASA	National Aeronautics and Space Administration
ODD	Operational design domain
OMG	Object management group
PHA	Preliminary hazard analysis
RAAML	Risk analysis and assessment modeling language
SLC	System-level constraint
SOTIF	Safety of the Intended Functionality

STAMP Systems theoretic accident modeling and processes

STPA System Theoretic Process Analysis

UCA Unsafe/undesired control action

5. DOCUMENT ORGANIZATION

This document is organized as follows:

- Basic STPA approach, recommended practices, lessons learned.
- High-level use of STPA within safety process and STPA with other safety evaluation methods.
- Summary.

6. BASIC STPA APPROACH AND LESSONS LEARNED

6.1 Introduction

This section describes attributes of a basic System Theoretic Process Analysis (STPA) approach. The content in this section comes from practitioner experience, and at times references, and excerpts from the STPA Handbook (Leveson and Thomas, 2018).

The STPA Handbook defines STPA as a hazard analysis technique based on an extended model of accident causation. In addition to component failures, STPA assumes that accidents can also be caused by unsafe interactions of system components, none of which may have failed. Some of the advantages of STPA over traditional hazard/risk analysis techniques are that:

- Very complex systems can be analyzed. “Unknown unknowns” that were previously only found in operations can be identified early in the development process and either eliminated or mitigated. Both intended and unintended functionality are handled.
- STPA can be started in early concept analysis to assist in identifying safety requirements and constraints. These can then be used to design safety (and security) into the system architecture and design, eliminating the costly rework involved when design flaws are identified late in development or during operations. As the design is refined and more detailed design decisions are made, STPA is refined to help make more and more detailed design decisions. Complete traceability from requirements to all system artifacts can be easily maintained, enhancing system maintainability and evolution.
- STPA includes software and human operators in the analysis, ensuring that the hazard analysis includes all potential causal factors in losses.
- STPA provides documentation of system functionality that is often missing or difficult to find in large, complex systems.
- STPA can easily be integrated into your system engineering process and into model-based system engineering.

STPA provides a system engineering and system thinking approach to the hazards analysis and initial requirements definition steps of system safety assessments. It provides a robust methodology to accomplish these tasks that is useful in any industry containing safety-critical systems. STPA effectively evaluates initial architectures and system concepts so appropriate tradeoffs and system balance discussions may occur or decisions may be made. STPA has been effectively used in many industries, and the approach outlined in this section is valid for any number of industries.

NOTE: This section does not intend to teach the STPA process, as other resources such as the STPA Handbook accomplish that goal. Please refer to these other resources for specific STPA questions. See Section 2 for additional resources.

6.2 STPA Method Overview

There are four steps in a basic STPA system safety evaluation. This section outlines these steps and provides recommendations to effectively execute these steps. The steps in a basic STPA are shown in Figure 1 (excerpt from STPA Handbook), along with a graphical representation of these steps.

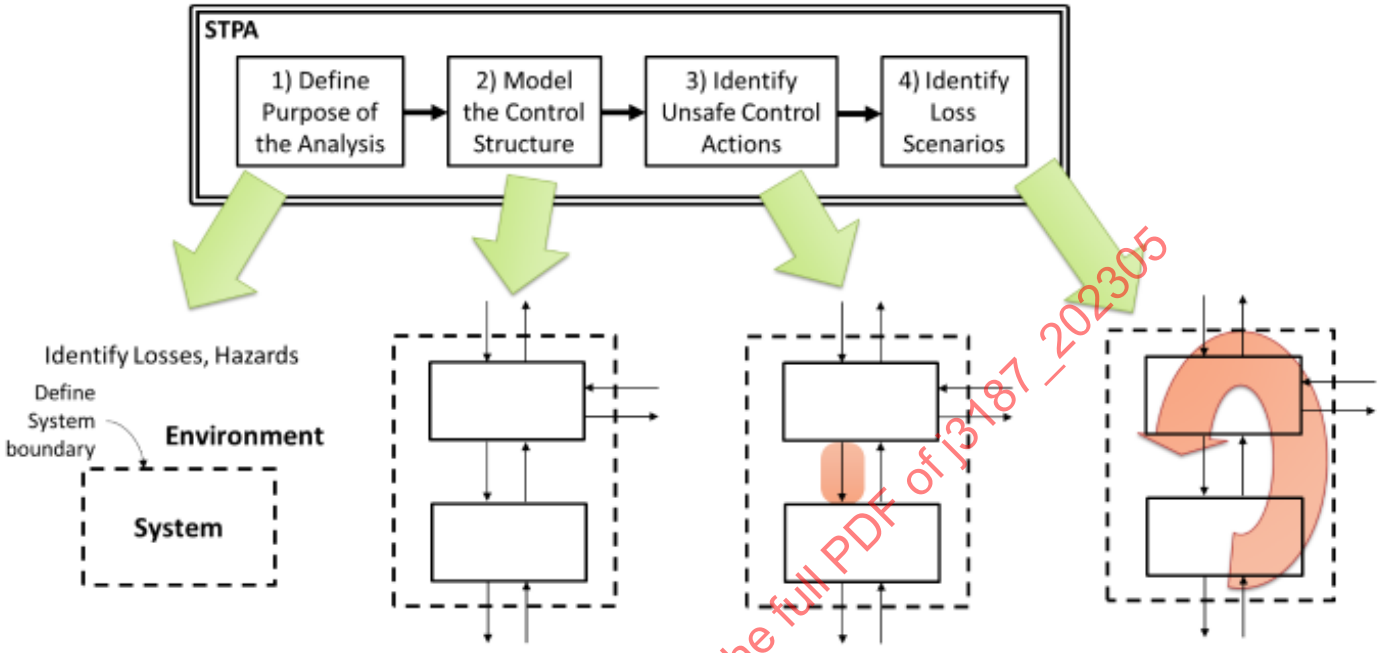

Figure 1 - Overview of the basic STPA method (Leveson and Thomas, 2018)
Used with permission from Leveson and Thomas. All other rights reserved.

The first step is to define the purpose and scope of the STPA evaluation effort. Accidents (losses) and safety goals associated with automotive applications are defined along with potential hazardous states (hazards) that may result from system element misbehaviors.

STPA can be used as a standalone process, or it can be used alongside existing frameworks. When used in an ISO 26262 framework, the STPA accidents and hazards are associated with potential harm to human operators, vehicle occupants, and surrounding pedestrians. (SAE J2980 is a good reference with examples of automotive hazard analysis.) When used in other frameworks, the STPA accidents and hazards can be linked to mishaps, hazards, or similar concepts.

High-level system constraints (safety goals) are developed by reformatting the hazards into constraints that will address the specific hazard. When used in an ISO 26262 framework, the STPA high-level system constraints can be listed as “safety goals,” which provides an approach to link safety requirements to safety goals and then to hazards[1]. The STPA high-level system constraints can be linked to safety objectives, high-level safety requirements, or similar concepts when used in other frameworks.

Building a hierarchical “control structure” is the second step. A control structure captures the functional interactions between system elements using a series of command and feedback loops. It is advantageous to model the system using a high-level control structure at an abstract level and then refine it to more detailed levels as the evaluation progresses. The control structure should not include implementation details but rather logical elements representing software controllers, actuators, sensors, and processes that the system controls. A control structure need not be limited to electromechanical elements of the system but may extend beyond the technical system to include elements such as human operators, maintenance personnel, a control room or fleet operations center, and interactions with the external environment.

[1] ISO 26262 defines a “safety goal” as a top-level safety requirement from the hazard analysis and risk assessment (HARA) at the vehicle level.

The third step is to identify unsafe or undesired control actions. This can involve examining functional responsibilities (functions) assigned to the various control structure elements. Undesired or unsafe control actions (UCAs) associated with each action that achieves a function are defined using standard syntax and then linked to the hazards defined in the first step. A UCA may lead to more than one hazard, and hazards may result from multiple UCAs (many-to-many relationships). It is worth noting that developing UCAs may help identify additional hazards not identified in the first step.

STPA is a systematic process to reduce and eliminate open-ended mental processes that result in gaps and oversights. Therefore, this step should avoid using non-systematic methods such as unstructured brainstorming.

The final step identifies the circumstances, conditions, or reasons (causal scenarios/causal faults) why UCAs might occur in the system. These scenarios are created to explain:

1. How unsafe or undesired control action leads to a loss event.

2. How incorrect feedback, inadequate requirements, design errors, component failures, and other factors can independently or collectively contribute to unsafe or undesired control actions ultimately lead to losses.

3. How safe (or desired) control actions, including hazard mitigation actions, might be required but are not commanded, are commanded improperly, or executed improperly.

More than one causal scenario can exist for each unsafe or undesired control action.

Causal scenarios create additional technical requirements, drive the architecture, and highlight conflicting requirements that must be resolved before the design can proceed.

If applied during early development, STPA can provide recommendations and new design decisions. It can also:

- Evaluate design decisions and highlight gaps when applied to existing concepts.
- Identify functional and behavioral requirements.
- Define human procedures and training requirements.
- Define test cases and create test plans.
- Develop leading indicators of risk.

6.3 Use of Systems Engineering Perspective

Effective safety analysis methods provide value through the entire engineering design, development, testing, and manufacturing phases of a system's implementation process. These analysis methods significantly impact system content in the early phases of a program as major expenditures for critical component procurements and manufacturing tooling have yet to be committed. There is much less cost associated with design changes at this point in the process than there are with changes that occur later in the process (e.g., it is much cheaper to move lines on paper than it is to move/change actual physical tooling lines in a manufacturing environment) (Vernacchia, 2018).

Early in the design process, various design concepts are explored to determine the optimized balance between system imperatives such as cost, timing, mass, performance, and safety "acceptable risk" requirements. This "system architecture balance" phase typically contains tasks such as those shown in Table 1, with some tasks having an iterative nature between them. Of course, if an optimal balance cannot be achieved, the iterative nature would be extended to early tasks to modify stakeholder expectations and initial requirements (Vernacchia, 2018).

Table 1 - System concept/architecture tasks involving early safety analysis methods (Vernacchia, 2018)
Used with permission from Vernacchia. All other rights reserved.

System Concept / Architecture Tasks	Safety Analysis Methods Areas
Stakeholder Expectation Determination	
Initial Requirements Identification	
Architecture Development / Modification	Iterative Loops Between these Tasks Until Optimum Balance Achieved
Balance, Trade Studies, Optimization Activities Leading to Design Selection within Functional Domain(s)	
Analysis Verification (Functional, Performance, and Safety)	
Program Validation Against Stakeholder Expectations	
Concept / Architecture Selection	

An underlying assumption in Table 1 is that system safety is an integrated part of the system engineering process to be optimally effective. The iterative loop shown in Table 1 may be further illustrated by a systems engineering "cascading conic" depiction shown in Figures 2, 3, 4, 5, and 6. Figure 2 shows the initial allocation of system imperatives to major systems. Figure 3 shows the interactions between these major systems during the balance, tradeoff, and optimization steps. Figure 4 shows the rebalance and optimization at the top system-level leading to an initial concept/architecture, and Figure 5 depicts the concept/architecture selection.

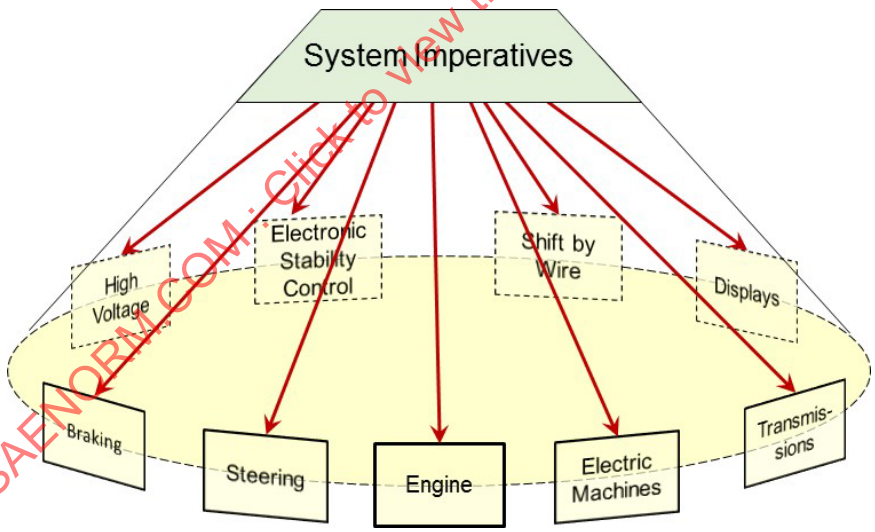

Figure 2 - Initial allocation to major systems (Vernacchia, 2018)
Used with permission from Vernacchia. All other rights reserved.

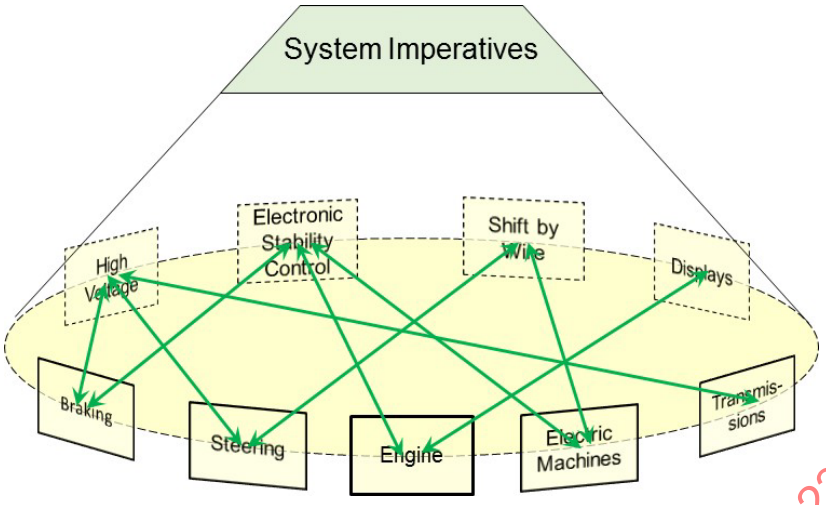

Figure 3 - Interactions between major systems (Vernacchia, 2018)
Used with permission from Vernacchia. All other rights reserved.

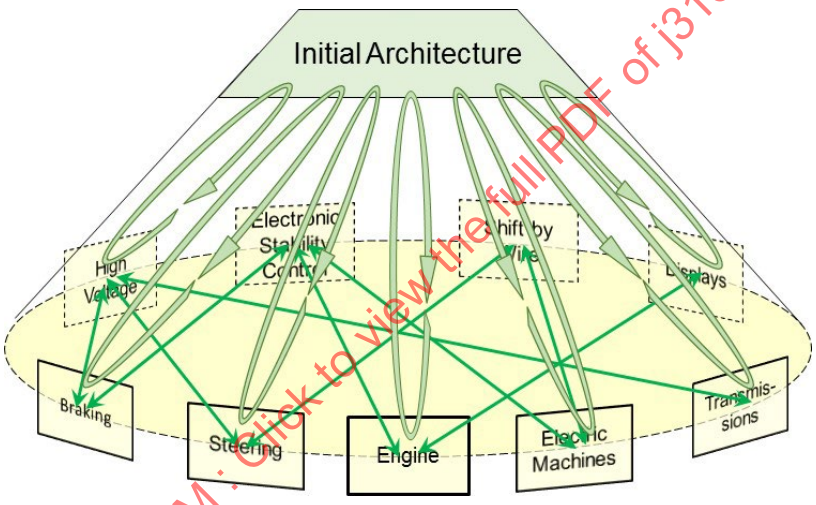

Figure 4 - Rebalance and optimize initial concept (Vernacchia, 2018)
Used with permission from Vernacchia. All other rights reserved.

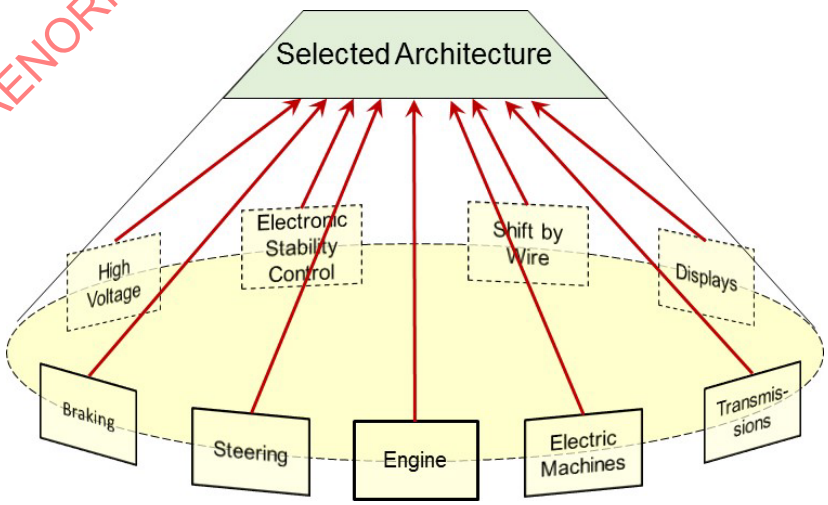

Figure 5 - Selected concept/architecture (Vernacchia, 2018)
Used with permission from Vernacchia. All other rights reserved.

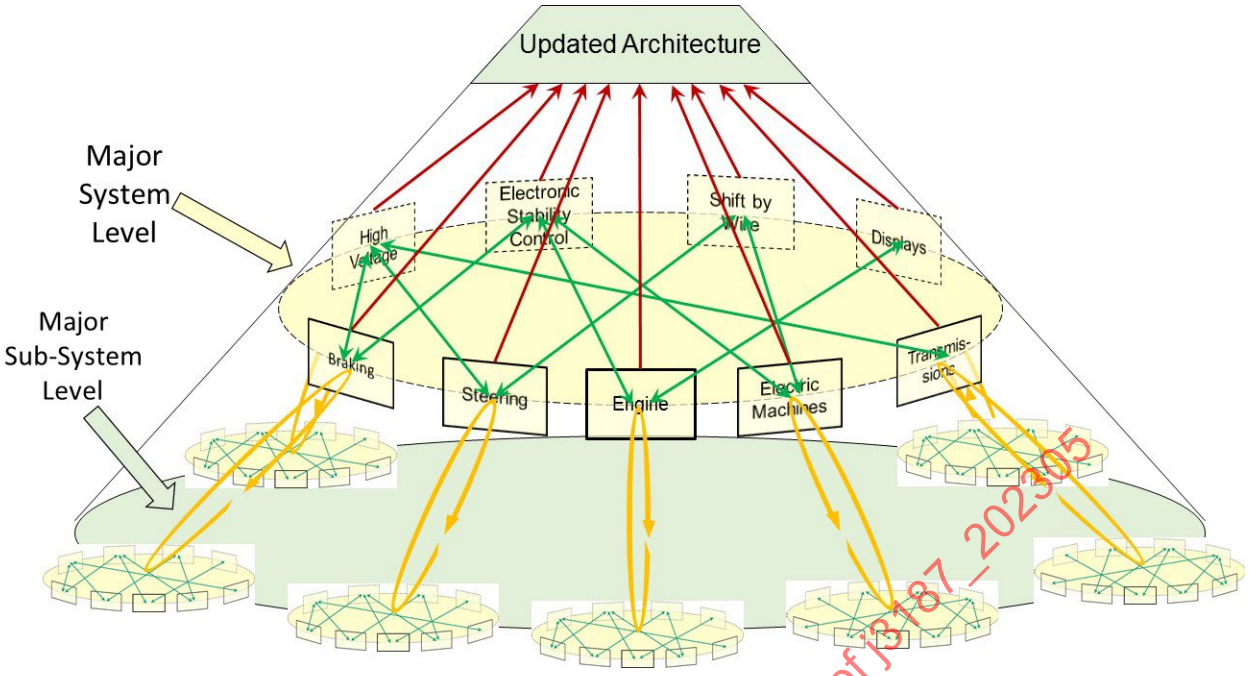

Figure 6 - Multiple level balancing and optimization for continued concept/architecture improvement (Vernacchia, 2018)
Used with permission from Vernacchia. All other rights reserved.

Figure 6 illustrates the allocation to the major subsystems of each major system where the balance, tradeoff, and optimization process repeats at this lower level, and updates are transposed back up to the top level. Two levels of evaluation tend to be sufficient for initial safety task activities. Deeper levels of the engineering process are very well served by familiar evaluation methodologies such as DFMEAs, FTAs, and FIAs.

Within this system's engineering process framework, safety tasks performed early in the design process support the balance and optimization tasks by identifying what hazardous conditions could arise due to system element misbehavior and assessing the risk associated with such hazards. This effort defines the required system content and the detection and mitigation strategies necessary to eliminate or manage the assigned risk to acceptable levels. To be effective, these detection and mitigation strategies must be part of the balancing and optimization of the system.

A key asset for this top-down, iterative, and interactive process is a "chief system architect" who understands the technical nature of the major system interactions and should have the expertise to broker tradeoffs and compromises between functional areas. If no such person is available, it is useful to engage "system architects" from the different functional domains and engage them in the "roll-down" and "roll-up" activities involving system imperative and high-level constraints.

6.4 STPA Scope Determination

The first step is to determine the system scope under consideration and its hierarchal nature. The scope of the STPA evaluation may change when going from the top hierarchal controller into a deep nested lower level controller, so the previous systems engineering-based section is an effective method to determine system content and element functions. Additional key considerations found effective to consider when defining the system are:

- Define the purpose of the analysis and the strategy/focus of the analysis. (For example, is this a safety analysis, security analysis, systems analysis, or some combination of the three? This purpose should be identified early on as it will lead to the identification and avoidance of losses, as mentioned in the next section.)

- Is safety-relevant security included in the analysis? Security's impact on safety needs to be considered from the beginning. Correlating security-related UCAs and scenarios with the system hazards and losses identifies the causal factors that have priority (Young, 2018).

- List all elements at the highest abstract level.

- The STPA control diagram is revisited throughout the analysis process, serving as a reference for the STPA evaluation and guiding discussions with design and test engineering teams.
- For these reasons, it is important to balance the level of abstraction and detail such that all aspects of the system can be analyzed across diverse teams.
- STPA should be applied initially at the highest abstract levels and then at more defined lower levels. A high level of abstraction is expected at the early definition stage, with details added as the project design progresses.
- Ask these upfront questions:

  A system to do . . . . . {What = Purpose}

  By means of. . . . . .{How = Method}

  In order to contribute to . . . . . .{Why = Goals}

  While . . . . . .{Constraints, restraints, ODD, etc.}

  System actors . . . . . .{Who or what is interacting with the system?}

  Delivered by . . . . . {When = Milestone deliveries}
- The key part is capturing system assumptions. If system assumptions are not detailed early in the system scope and boundary, difficulty will develop when trying to understand causal analysis and process model flaws. So, capture all assumptions of system behavior early on. Examples: “This analysis will only focus on automatic transmissions,” or “This analysis is assumed to have Level 3 autonomy only.”

Additional items to consider while determining scope and boundary are:

- List assumptions as to the basis for focused analysis.
- Think of different ways to frame the problem.
- Avoid considering only the items we control, but intentionally look at all aspects with a specific interest in items that are not controllable (or perceived not to be controllable).
- Must ensure that we define and solve the right (engineering) problem.
- Group list of elements but avoid detailing lower levels and instead focus on top-level interactions of an element. It is okay to show some dependencies, but it is better to focus on top-level understanding.
- Consider not just communication but process interfaces (e.g., high voltage for electric powertrain).
- Identifies the system boundaries as well as defines the components (subsystems) inside the system boundaries targeted by STPA.
- Definitions of the components inside the system boundaries to the granularity of the control structure.

6.5 Potential Loss (“Accident”) Identification

The next step is to identify potential accidents (e.g., “loss,” “harm”) that may occur if certain hazardous states and specific circumstances are present. In STPA, a “loss” can occur due to an accident. ISO 26262 uses “harm” as an equivalent for an accident. This document will use each of the three words—accident, loss, and harm—as equivalent terms.

STPA can be used to target any loss that is unacceptable to stakeholders. If more than one loss is included, they can be ranked and prioritized. Because every STPA result will be traceable to one or more losses, the analysis results can be easily ranked and prioritized based on the losses to which they refer.

Before any analysis begins, the stakeholders must identify the losses they want the analysis to focus on. The losses to be considered may be defined by management, government regulations, or customers. A general approach to identifying losses may involve:

1. Identify the stakeholders, e.g., users, producers, customers, operators, etc.

2. Stakeholders identify their “stake” in the system. What do they value? For example, human life, a fleet of automobiles, electrical power generation, transportation, etc. What are their goals? For example, to maintain a fleet of automobiles, provide transportation, provide medical treatment, provide electrical power generation, etc.

3. Translate each value or goal into a loss, e.g., loss of life, loss of vehicle functions, loss of electrical power generation, loss of transportation, etc.

As an example, automotive application accidents will typically fall into one of the following categories:

- Vehicle collides with another vehicle.
- Vehicle collides with pedestrian(s).
- Vehicle occupant injury.
- Vehicle collides with infrastructure.

In addition to these categories, accidents (i.e., harm) result from other occurrences, such as a fire or an electrical shock that may occur without a vehicle collision.

Depending on the purpose and scope of the STPA effort, the evaluation of accidents (losses) can expand beyond safety-related incidents. Accidents can include various other types of losses, such as performance, financial, quality, risks, and threats. Accidents do not have to be limited to safety-related losses and should include other important design aspects as appropriate. A hazard need not always lead to a loss but may lead to a loss under certain conditions. For example, unintended braking may be a hazard for automated emergency braking (AEB), but it may lead to a loss if there is a following vehicle that collides with the subject vehicle.

Additional key aspects to consider when identifying accidents and losses are:

- Use top-level strategy and priorities to focus on consequences directly related to the strategy, purpose, and definition of the system being analyzed.
- Verbs may be used to help identify system accidents and hazards:
  - Autonomous vehicle (AV) colliding with mobile object.
  - Autonomous vehicle (AV) injuring passengers without a collision.
- If security is included in the analysis, distinguish between safety-related accidents and non-safety-related security losses, such as loss of reputation, intellectual property, privacy or breach of customer personal information. Non-safety-related losses can be related to quality, performance, security, and financial.
- For traceability, use unique labels for each accident and loss (security loss not impacting safety).

Next are potential hazardous situations (system-level hazards) that may lead to accidents/losses in specific circumstances that need to be determined.

6.6 Potential “Hazard” Identification

Hazards should first be identified at the system level, i.e., associated with the high-level control structure representation. (More hazards may be defined as more detailed control structures are evaluated.) High-level control structure hazards for vehicle applications (e.g., road vehicles, aerial vehicles, space vehicles, etc.) might include vehicles exceeding safe operating envelope, vehicles not maintaining a safe distance from nearby objects, or vehicles entering dangerous areas/regions. These hazards may lead to multiple accidents and losses. As such, the STPA process links each hazard to the resulting accidents and losses by typically documenting the accident/loss identifier in brackets after the hazard.[2]

The following criteria from the STPA Handbook (Leveson and Thomas, 2018) is useful for defining hazards:

- Hazards are system states or conditions (not component-level causes or environmental states).
- Hazards lead to an accident (or loss) in some worst-case environment.
- Hazards must describe states or conditions to be prevented.

The STPA Handbook goes on to outline some common mistakes when identifying hazards:

- Confusing hazards with causes of hazards.
- Too many hazards containing unnecessary detail.
- Ambiguous or recursive wording.
- Confusing hazards with failures.
- Key pitfalls: If you use words like “causing” or “resulting in,” you are likely combining a hazard and a potential cause. See the STPA Handbook for examples.

It may be helpful to list out a few possible mechanisms for developing the initial hazard list before entering the UCAs (e.g., field data). The link here shows a high-level HAZOP to help find this initial list.

System-level hazards will be a list that will evolve as the STPA process is implemented, especially as the control diagram(s) are refined. Start with the list of elements created for a top-level understanding of the system. Use losses that were identified earlier to identify related hazards. Use verbs to identify hazards and avoid compound sentences. For example:

- AV breaches the minimum safe distance of a forward mobile object.
- AV exposes passengers to unhealthy g-forces.

If security is included in the analysis, distinguish between hazards with the potential cause of an accident and losses that don’t impact safety. Listing both safety and non-safety related losses adds clarity to the boundary of the analysis. As the analysis proceeds, it is possible that elements could move to the other list. It is important to look across the analysis boundary to leverage the iterative nature of STPA.

For traceability, use unique labels for each hazard and loss.

---

[2] In ISO 26262, “accidents” are not specifically linked to hazards. ISO 26262 investigates when hazardous situations may lead to “harm” under certain operating conditions or modes. An accident is an example of harm. Even with this distinction, ISO 26262 links hazards to an accident (harm) by hazard analysis and risk assessment (HARA). A system item is evaluated without internal safety mechanisms for various operational situations and operating modes. An operational situation or mode may define a hazardous event, which, unmitigated, may lead to a loss.

ISO 26262 provides one way to assess the impact of a potential loss using three parameters: severity, exposure, and controllability. Each of these parameters has multiple levels of degree and then can be combined to define an automotive safety integrity level (ASIL) rating. The five ISO 26262 classifications are (starting with most significant) ASIL D, ASIL C, ASIL B, ASIL A, and QM. QM (quality managed) is the least significant classification and allows a robust engineering process to address system safety concerns.

6.7 Defining System-Level Constraints (Safety Goals)

The definition of STPA system-level constraints follows the hazard identification step in the STPA process. In this document, the term “safety goals” will be used.[3]

STPA offers a simple way to identify these system-level constraints by taking the hazard description and inverting the condition. A suggested format is provided here.

Hazard Listing:

Hazard Identifier -

System of Interest -

Unwanted/Unsafe Condition -

Association to Accidents/Losses

System-Level Constraint (SLC):

SLC Identifier -

System of Interest -

Constraint to Implement -

Hazards Prevented

Using this format, two examples are presented here.

Hazard-1: Vehicle experiences an unwanted acceleration causing the driver to lose control (may lead to an accident where a collision with another vehicle or injury to occupant(s) or pedestrian(s)).

Safety Goal-1: Vehicle shall not experience an unwanted acceleration to a level that causes a driver to lose control where collision or injury could occur.

Hazard-2: High voltage battery causes electric shock (which may lead to an accident where humans are injured).

Safety Goal-2: High voltage system shall not allow an electric shock to injure humans.

These constraints are high level and, as such, are considered “safety goals.” Such safety goals will be assigned more detailed safety requirements as the system design develops. Having such safety goals enables more understandable traceability of requirements to hazards to accidents (losses).

[3] ISO 26262 defines a “safety goal” as a top-level safety requirement resulting from hazard analysis and risk assessment (HARA). Non-safety-related losses, top-level security goals, and general system goals can also be created to link to associated risks and threats.

Once the system-level hazards are identified, it is straightforward to identify system-level constraints that must be enforced: simply invert the condition.

<Hazard> = <System> & <Unsafe Condition> & <Link to Losses>
<Safety Constraint> = <System> & <Condition to Enforce> & <Link to Hazards>

H-1: Aircraft violate minimum separation standards [L-1, L-2, L-4, L-5]
SC-1: Aircraft must satisfy minimum separation standards from other aircraft and objects [H-1]

H-2: Aircraft airframe integrity is lost [L-1, L-2, L-4, L-5]
SC-2: Aircraft airframe integrity must be maintained under worst-case conditions [H-2]

Each constraint can be traceable to one or more hazards, and each hazard is traceable to one or more losses. In general, the traceability need not be one-to-one; a single constraint might be used to prevent more than one hazard, multiple constraints may be related to a single hazard, and each hazard could lead to one or more losses.

Figure 7 - STPA Handbook system-level constraints (safety goals) (Leveson and Thomas, 2018)
Used with permission from Leveson and Thomas. All other rights reserved.

6.8 Creating a Control Structure

A control structure is a representation of the “control” hierarchy of the system. The controls hierarchy of a control structure typically starts with a three-level structure showing the “system” being controlled at the lowest level, the “controller” driving that system at the mid-level, and the ultimate element commanding the controller at the highest level. An example would be the operator at the top of the hierarchy, a controller/computer interacting in the middle, and the system being controlled at the bottom. Figure 8 shows such a control structure example.

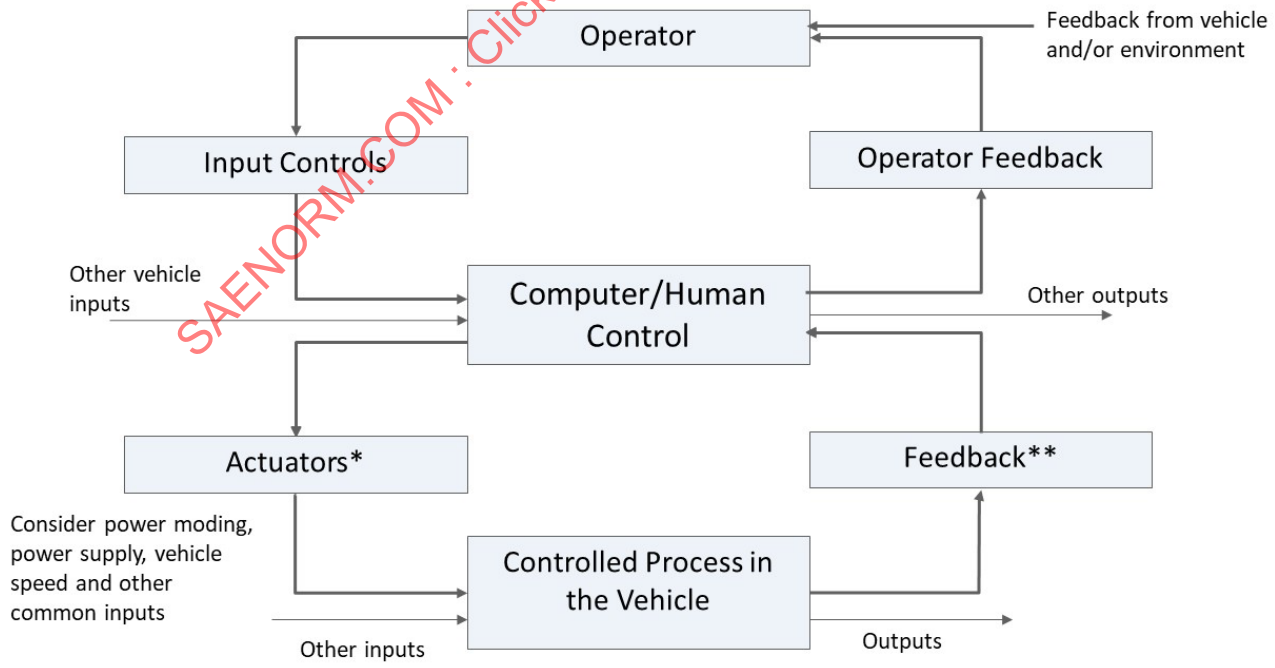

Figure 8 - High-level control structure

Vertical hierarchy denotes elements that are shown “higher” (towards the top) in the control structure and have higher precedence of decision-making responsibilities. As such, the hierarchy indicates control authority between the system elements. Therefore, an element has control over the elements below it and is controlled by elements immediately above.

In addition to hierarchy, Figure 8 illustrates the control loops between system elements represented by the arrows. STPA notation convention uses downward arrows to convey control actions or commands and upward arrows to convey feedback.

The control structure representation of a system should begin at the highest level possible. This enables the use of abstraction to manage complexity. A dictionary definition of abstraction is “the process of taking away or removing characteristics from something to reduce it to a set of essential characteristics.” This approach allows the STPA practitioner to develop multi-level control structures with each lower level, illustrating the next level of complexity in the system. Figure 9 shows a progression of abstraction, for example, compressed natural gas (CNG) propulsion systems.

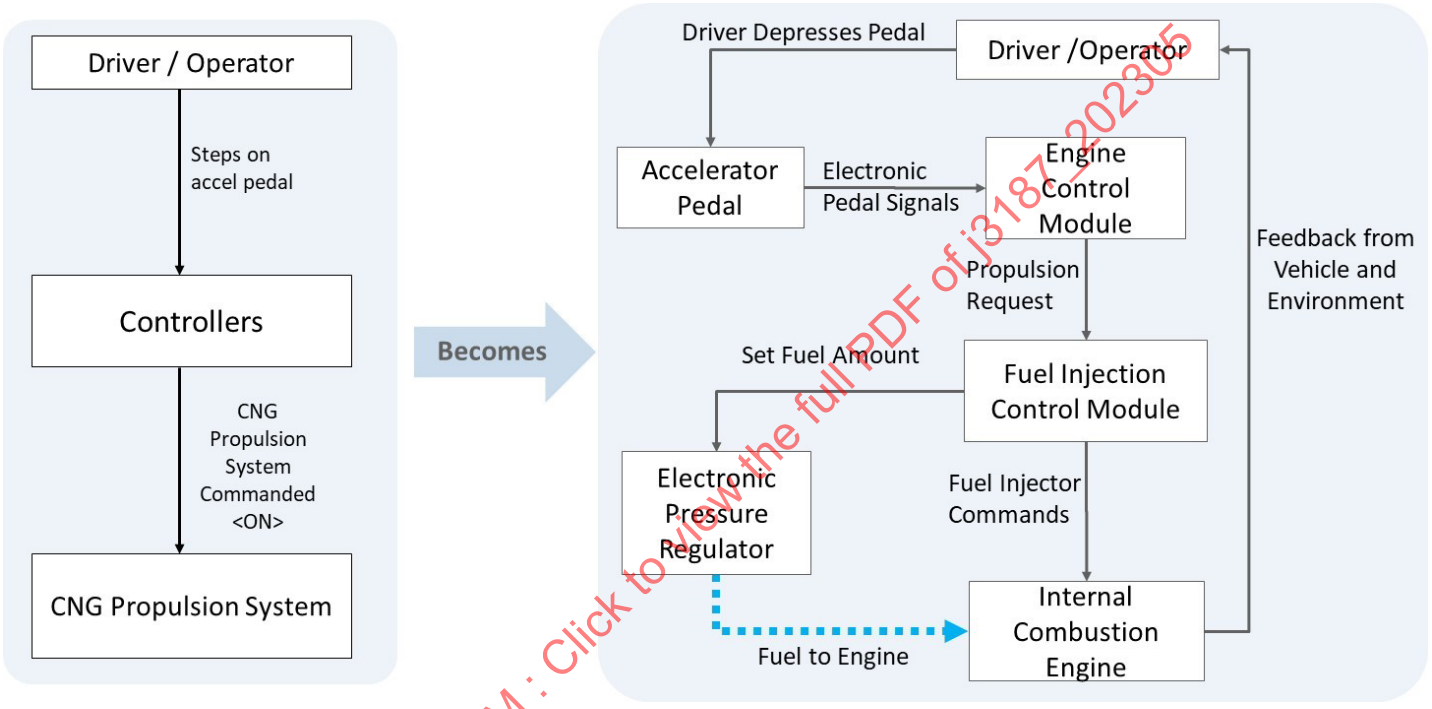

Figure 9 - Abstraction progression example

Once the control structure abstraction level is decided, the next step would be to assign functions to the control structure elements. These functions (aka “responsibilities”) may be achieved using command actions from elements, feedback activities from elements, and/or extensions of the system-level constraints. Each element should understand what it must do to be sure the safety goals are achieved. At times, these functions may also describe what the element is to do to enable the system to provide safe behaviors.

Abstraction is an STPA-recommended approach to manage the complexity of the analysis. It allows higher-level representations of UCAs so that the evaluation will not bog down in numerous details too early in the process. This enables a practitioner to manage the complexity by deferring the detailed level analysis to later portions of the evaluation. In addition, it enables the identification of high-level requirements that would manage lower-level detailed requirements. Ultimately, the final requirements seem to be the same regardless of abstraction. Still, the non-abstraction approach risks the evaluation collapsing as the effort required to complete the analysis appears too high.

Abstraction is also useful where numerous causal scenarios drive many requirements. A practitioner may observe that the same requirement addresses many causal scenarios.

Each analysis in a determined level of abstraction will link several activities. Evaluations of control structure elements lead to identifying UCAs, which then lead to determining causal scenarios, which finally generate safety requirements or constraints. These safety requirements lead to design decisions affecting the architecture, so functions/control actions are allocated to human/HW/SW. This leads to the construction of a more detailed functional control structure that leads to new UCAs, etc., as shown in Figure 10.

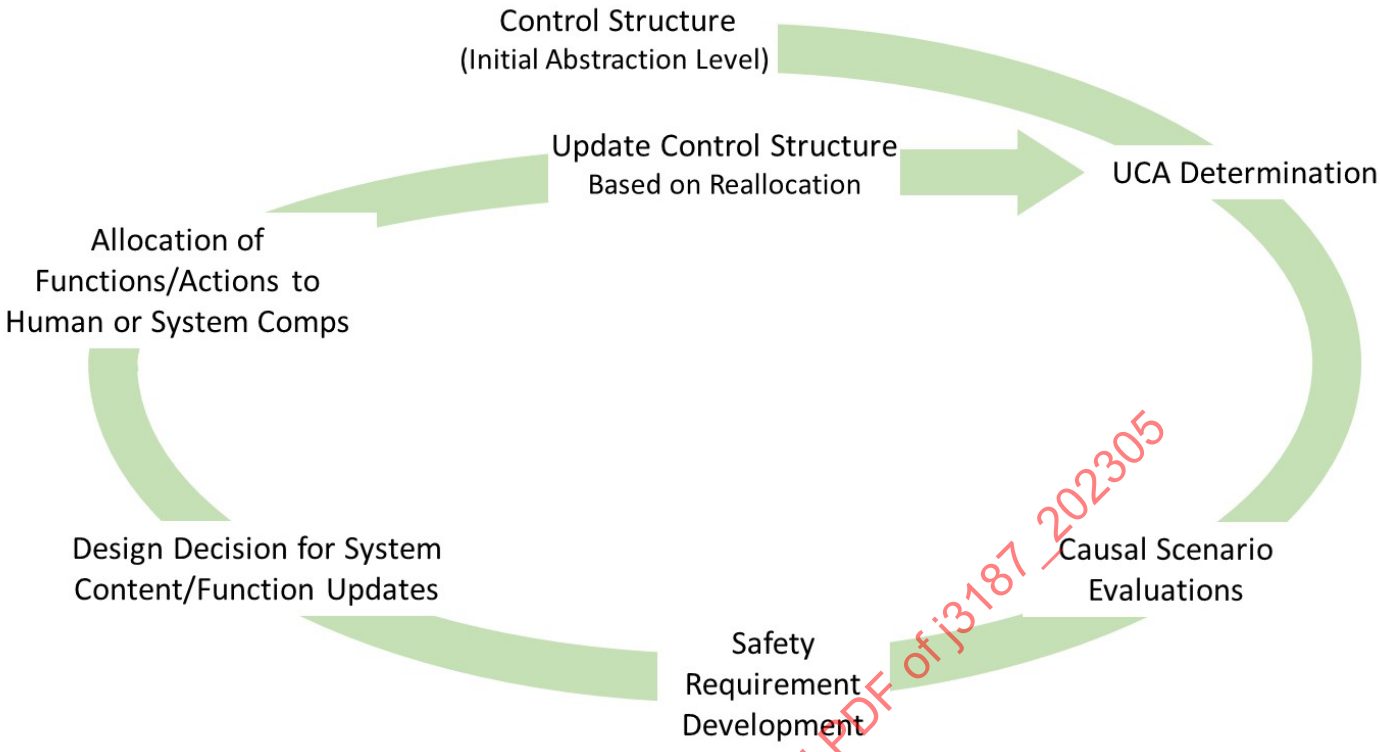

Figure 10 - Iterative cycle using STPA for system control structure updates

Therefore, a sequence with the following structure is constructed to maintain traceability:

UCA → Scenario → Safety Requirement → Design Decision →
Allocation of functions/control actions to human or system components →
Traceability to a new functional control structure.

Ideally, STPA begins with the start of the system conceptualization. When STPA starts after system design has already started, a decision needs to be made to either:

- Incorporate existing design details into the analysis, using analysis results to impact design change retroactively at whatever stage of the design it is in; or
- Ignore existing design/implementation details initially, first implementing an abstract analysis independent of the detailed system design.

Once STPA is done, a gap analysis can be implemented for the existing design and any other architecture designed for the same purpose. The advantage of this approach allows for the STPA to be reused for multiple architectures.

The control diagram is expanded hierarchically by adding detail to each sub-level starting from the top-level. This approach facilitates understanding the system and related elements.

6.9 Define Unsafe/Unwanted/Unexpected Control Actions (UCAs)

An unsafe control action (UCA) is a control action that will lead to a hazard in a particular context and worst-case environment. The term “unsafe” refers to the hazards identified in STPA. As discussed earlier, hazards can include issues related to the loss of human life or injury (traditional safety). Still, losses can also be defined much more broadly to include other losses like mission loss, loss of performance, environmental losses, and more (Leveson and Thomas, 2018). Events leading to non-safety related losses could be documented as a risk and considered a type of hazard.

STPA identifies unsafe control actions that may lead to hazards. The STPA Handbook (Leveson and Thomas, 2018) states that there are four ways a control action can be unsafe:

1. Not providing control action leads to a hazard.
2. Providing control action leads to a hazard.
3. Providing a potentially safe control action but too early, too late, or in the wrong order.
4. The control action lasts too long or is stopped too soon (for continuous control actions, not discrete ones).

These four ways are organized into a tabular heading and each category is analyzed:

Control Action	Not providing causes hazard	Providing causes hazard	Too early, too late, out of order	Stopped too soon, applied too long

Figure 11 - Guideword headings as shown in STPA Handbook (Leveson and Thomas, 2018)
Used with permission from Leveson and Thomas. All other rights reserved.

Categories are useful to help identify the depth of the analysis. Sub-categories are possible. For example, “providing causes hazard” can include subcategories like “control action is intermittent,” “control action is too high,” and “control action is too low.” The STPA Handbook (Leveson and Thomas, 2018) contains additional guidance about these categories and subcategories.

Some control actions may directly represent a function of the system element, like braking. In that case, the control actions can be analyzed as identified “functions” of the system element of interest defined when developing the control structure content. In addition, STPA practitioners have made some enhancements by providing some helpful phrases under each guideword column, looking like Figure 12.

	"NOT Providing" Cause Hazard	"Providing" Cause Hazard	Incorrect Timing Incorrect Order	Stopped Too Soon Applied Too Long
Function (Responsibility)	Missing Not Followed	Providing When Not Expected Provided More/Less than Required Provided Content Results in Contol Conflict	Provided Too Early/Late When Required Provided Before/After When Required Provided Content in Wrong Order Provided Opposite of What Expected	Providing Unstable or Osscilating Content Providing Truncated Content Providing Stuck Content

Figure 12 - Guideword table with additional helpful phrases (Ressler and Vernacchia, 2019)
Used with permission from Ressler and Vernacchia. All other rights reserved.

Because the UCAs are used to define when behavior should or should not be required, the UCAs cannot assume that the requirements are known, or if they are known, they must not assume that they are correct or complete. The same is true for functions - the UCAs cannot assume that the required or intended functions are safe. A UCA is not limited to compliance with designed, intended, or required behaviors. A UCA can describe any behaviors that lead to hazards, including designed, intended, and required behaviors.

A UCA may not simply state “Controller X does not provide brake command when intended” or “Controller X provides brake command when not expected” because that assumes the intended and expected behaviors are always safe. Instead, the UCA context must define the underlying conditions that make the action unsafe without relying on requirements, intentions, and expectations. An example of a stronger UCA is: “Controller X does not provide brake command when a forward collision is imminent.” This UCA does not just rely on assumptions about intent, expectation, or existing requirements but instead describes the underlying condition (imminent forward collision) that makes the lack of control action unsafe.

UCA format is a key factor for conducting an effective STPA evaluation. UCAs should contain five different parts. The first part is the “source,” the system element that produces the control action. The second part is the “type” of UCA that corresponds to one of the guide word column headings. The third part is the “control action” itself. The fourth part is the “context,” in which the control action is produced. And the final part is the “link” to the identified hazard(s). An example of a function that will provide hill hold capability when the vehicle stops on an incline and where the hazard would be unintended vehicle motion is shown in Figure 13.

NOTE: BSCU is brake system control unit in Figure 13.

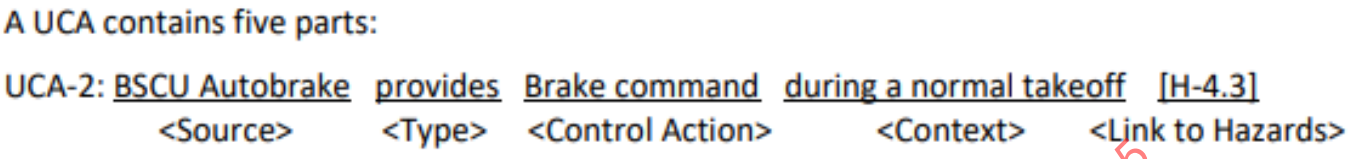

Figure 13 - Example of appropriate UCA format
Used with permission from Leveson and Thomas. All other rights reserved.

The STPA Handbook will note that “UCAs are often written with each part in the same order shown above, but in some cases, it may be clearer or more natural to use a different ordering. The ordering is not critical. The key point is that UCAs contain these parts.”

UCA development may become expansive for complex systems as more and more abstraction levels are evaluated. An STPA practitioner may be faced with either creating numerous specific UCAs according to all the possible operating contexts or creating fewer higher-level UCAs to accommodate several similar use cases. While either will support the downstream development safety requirements based on causal scenario evaluations, the second approach is more effective in handling the prospect of having so many specific UCAs at the beginning of the evaluation that the problem seems too daunting to approach.

The analysis should link each newly identified UCA to either a hazard, a risk, or both. This will help prioritize the safety-related UCAs for the analysis effort and keep track of risks that require system requirements. Ultimately, the goal is to generate good safety and system-level requirements that should not conflict with one another. In the event the analysis reveals a conflict between a high-level system requirement and safety requirement, an engineering management decision may be warranted on how to proceed.

Examine each UCA category for each element’s control action or function to generate UCAs.

- Identify if there is a need to iterate between UCAs and hazards.
    - Each UCA must be associated with a hazard. If a UCA is identified but no appropriate hazard exists, this may indicate that a new hazard is needed in the analysis. In this sense, hazard identification and derivation of UCAs is an iterative process.
- Deriving the context for UCAs can be done systematically by identifying key high-level inputs necessary for the control action or function. For instance, this might include the operator’s input to the system and inputs from interfacing systems. The control action/function and UCA category could assess each combination of inputs.
- Remove UCAs not associated with any hazard.
    - Highlight those UCAs for further STPA evaluation to identify constraints for risks/threats and other losses to which they could be linked.
    - Some UCAs could be related to overall system performance or the intended mission, so they should not be removed, just linked to other hazards/losses and highlighted for further analysis.
- Develop constraints for UCAs using “not” logic.

- Identify UCAs through a systematic reference of every control signal in the control diagram, using the four STPA criteria for these signals:
    - Safe action not provided.
    - Unsafe action provided.
    - Incorrect timing or order.
    - The action stopped too soon.
- Reference system strategy/focus to identify UCAs with the most consequence.
- Standardize words and phrases used to define CAs and UCAs, expecting duplication and looking for automation opportunities with available STPA tools. Consider using editing tools such as LaTeX to program standard phrases. Standardizing phrases helps with automation and reduces error and adds clarity for finding unique CAs.
- For traceability, use unique labels for each UCA.
- Make sure the control diagram captures all CAs. For instance, if the driver has a CA to “depress the accelerator pedal,” the analyst should also consider CAs to “release the accelerator pedal.”
- Expect to refine the control diagram as UCAs are determined, and additional signals and/or elements are identified.

As UCAs are determined and additional signals and/or elements are identified, expect to refine the control diagram.

6.10 Define Casual Scenarios

This step explores the possible causal scenarios that could lead to the identified unsafe control actions. Information is generated to assist designers in eliminating or mitigating the potential causal scenarios of the hazard. This involves examining the control loops and their parts and identifying how they could lead to undesired control action.

Causal scenarios are not faulted states like those in FMEAs, although the most basic causal scenarios may be similar to faults identified through FMEA. In FMEAs, known failures are evaluated, and their effects are assessed, so the key question is “what” failed. There may not be any failure in causal scenarios but rather a set of conditions or circumstances that could lead to a situation where a UCA might occur.

The key question in causal scenario evaluation is to ask “why.” This leads to two types of questions: first, “why” would UCAs occur; second, “why” would control actions be improperly executed or not executed, leading to a hazard. The STPA Handbook summarizes this point as follows. It proves a graphic (Figure 2.17 from the Handbook, as shown in Figure 14) illustrating the application of these scenarios to a control structure (Leveson and Thomas, 2018).

Two types of loss scenarios must be considered:

a. Why would unsafe control actions occur?

b. Why would control actions be improperly executed or not executed, leading to hazards?

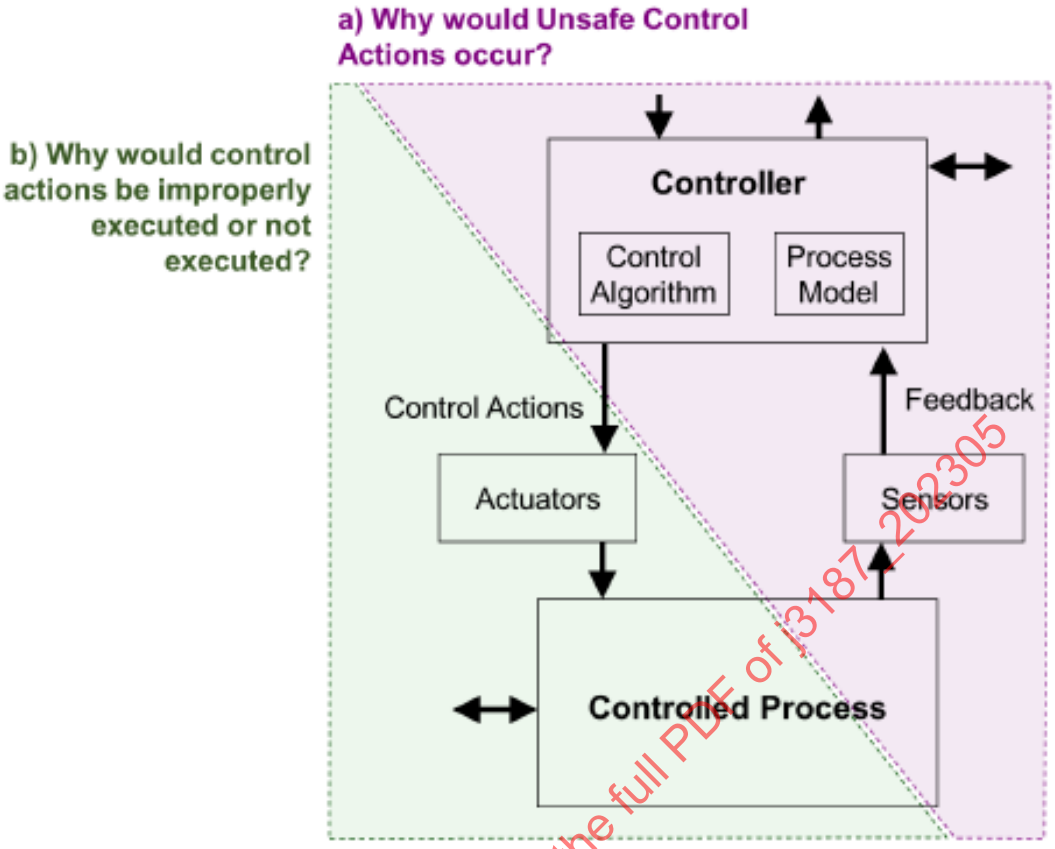

Figure 14 - Causal scenario graphic from STPA Handbook (Leveson and Thomas, 2018)
Used with permission from Leveson and Thomas. All other rights reserved.

The two types of loss scenarios can be expanded by leveraging the previously mentioned “why” aspect of the causal scenario evaluation and breaking up the control structure into four areas. These areas are listed here, with examples provided.

- Inappropriate decisions—Type (a) from above:
  - Controller micro-processor issues.
  - Poorly designed algorithms.
  - System changes that make the process model obsolete.
- Inadequate feedback and other inputs—Type (a) from above:
  - The sampling frequency of sensors is inadequate.
  - Conflicting feedback inputs.
  - Incorrect feedback.
- Inadequate control execution—Type (b) from above:
  - Actuators receive conflicting commands from multiple controllers.
  - Actuators do not execute a command.
  - Actuator response time is inadequate.

- Inadequate process behavior—Type (b) from above:
  - The controlled process has other actuators from other systems affecting it.
  - The controller process does not respond to the actuator action.
  - The controller process does not execute the control action correctly.
  - Controlled process missing process element (e.g., high voltage power for electric powertrain).

Figure 15 shows an example of a compressed natural gas (CNG) propulsion system control structure with shaded areas depicting the two-major causal scenario type coverages. This CNG system can be powered by either natural gas or regular gasoline. The fuel injector control module (FICM) controls the flow of high-pressure CNG through a shutoff valve and then through an electronic pressure regulator that drops the CNG pressure to a level where the system fuel injectors can administer the fuel into the internal combustion engine. Fuel tank and fuel rail pressure sensors provide feedback to the FICM.

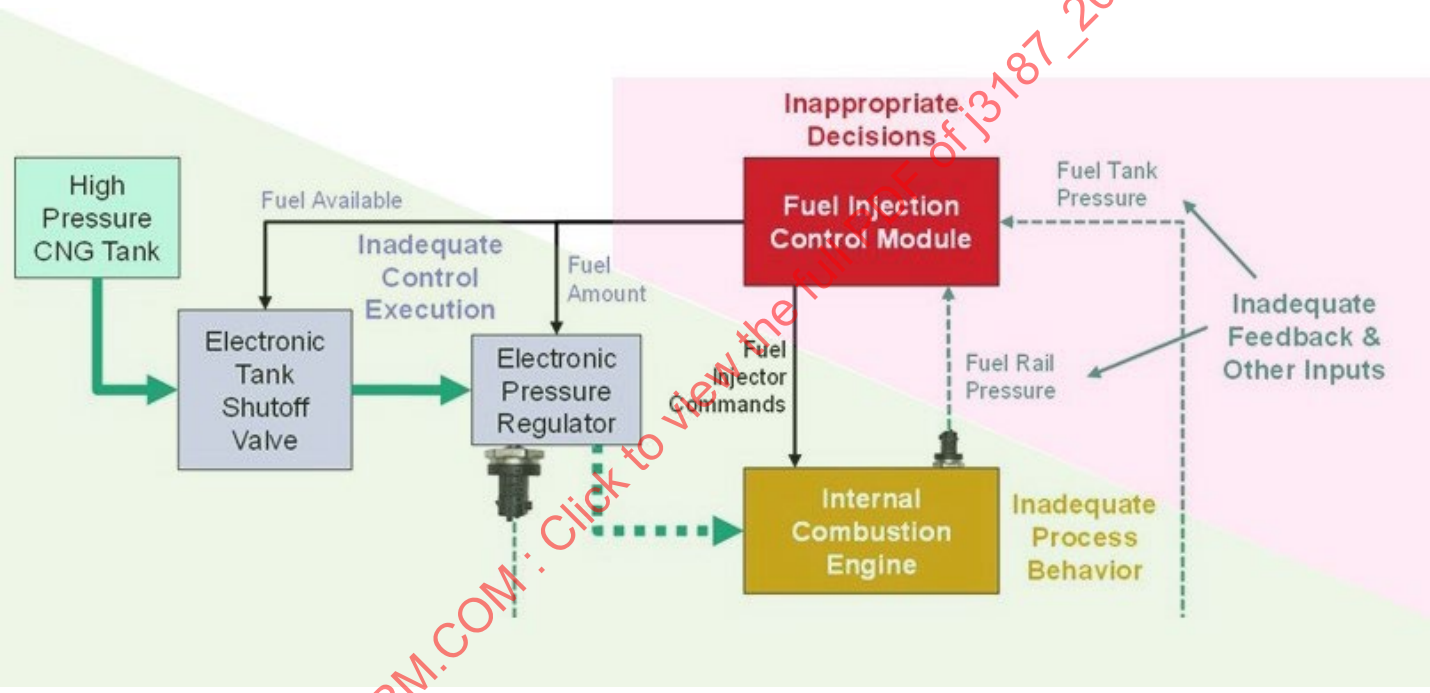

Figure 15 - Compressed natural gas system: causal scenario "type" areas (Ressler and Vernacchia, 2018)

Inspection of the four areas mentioned above could lead to the identification of causes such as:

- Inappropriate decisions (fuel injection command module):
  - FICM did not realize the system was in gasoline mode.
  - FICM was designed to wait for driver input to switch fuel modes.
- Inadequate feedback and other inputs (sensors):
  - Low voltage to fuel tank pressure sensor.
  - Fuel tank pressure sensor resolution inadequate.

- Inadequate control execution (actuators):
  - fuel rails and injectors are over-pressurized.
  - Pressure regulator response time too slow.
  - Shutoff valve does not close completely.
- Inadequate process behavior (CNG engine):
  - Fuel injectors do not provide appropriate pressure.
  - Fuel injectors provide CNG when in gasoline mode.

These causes would appear in causal scenarios explaining how one or more causes come together, leading to hazard, and ultimately losses. Once causal scenarios have been evaluated, the STPA process identifies the safety requirements necessary to prevent or manage these causal scenarios to an acceptable risk level.

Additional guidance for defining causal scenarios and causal factors includes:

- Define causal scenarios.
  - Identify scenarios that could lead to UCAs. (Why would UCAs occur?)
    - Bad “controller” behavior.
    - Inadequate feedback/information causes.
    - The key part is capturing system assumptions and ways they could be violated. If system assumptions are not detailed early on in the system scope and boundary, it will become apparent when trying to understand causal analysis and process model flaws, so capture all assumptions of system behavior early on.
  - Assess why commands/instructions (control actions) would have problems being conducted or not performed.
    - Control path issues.
    - Controller process entity issue.
  - For implementation, there needs to be a systematic process using STPA provided criteria to test every signal in the control diagram if it is: missing, inadequate, incorrect, or delayed. “S” denotes “scenario.”
    - S1 = AV breaches minimum safe distance of the forward mobile object when braking force command not provided in time.
    - S2 = AV exposes passengers to unhealthy g-forces when incorrect braking force command is provided.
  - Identify safety goals and UCAs constraints and perform requirements gathering.
    - Using a parameterized approach to developing context for UCAs can help identify parameter combinations that lead to conflicts that need to be resolved through system design (e.g., not providing an action may lead to hazard H1, but providing the action may lead to a different hazard).
  - Develop top-level requirements—the system should not do “X” under these conditions.
  - Incorporation of safety requirements into appropriate technical documents for system content design, supplier-provided entities, and validation test cases and planning.

- Define causal factors.
  - For each scenario, consider behaviors throughout the control diagram, such as: controller (group 1), actuator (group 2), controlled process (group 3), and sensors (group 4).
  - For S2 example:
    - Controller (group 1): Incorrect braking force due to a compute HW fault, threshold value memory fault, etc.
    - Actuator (group 2): None.
    - Controlled process (group 3): None.
    - Sensors (group 4): None.
  - Take a systematic approach for each group.
  - Provide a “rationale table” for each scenario analyzed for causal factors. This is critical to capture the “why” of the analysis for later reference.
  - Use causal factors to refine control constraints—iterative STPA process.
  - Develop top-level requirements—the system should not do “X” under these conditions.
  - Incorporation of safety requirements into appropriate technical documents for system content design, supplier-provided entities, and validation test cases and planning.

6.11 Creation of Safety Requirements

Safety requirements ensure the system can prevent or manage identified causal scenarios to an acceptable risk level. Each causal scenario is evaluated over its potential operating conditions, and appropriate requirements are developed. An example of this effort was developed using the causes from the CNG system in the last section, where the requirements are shown in italics text (Ressler and Vernacchia, 2018). (Used with permission from Ressler and Vernacchia. All other rights reserved.)

- Inappropriate decisions (FICM):
  - FICM did not realize the system was in gasoline mode.
  - FICM was designed to wait for driver input to switch fuel modes.
    - *ECM shall send a secured request to FICM indicating GAS mode is active.*
    - *FICM shall not activate CNG mode if the ECM signal indicates a gas mode.*
    - *FICM shall automatically switch fuel modes when the existing mode is unavailable.*
- Inadequate feedback and other inputs (sensors):
  - Low voltage to fuel tank pressure sensor.
  - Fuel tank pressure sensor resolution inadequate.
    - *Fuel tank pressure (FTP) sensor voltage supply shall be monitored by FICM.*
    - *FICM shall disable CNG operation if the FTP sensor is not available.*
    - *FTP sensor resolution shall be 3 psi over the full operating range.*

- Inadequate control execution (actuators):
  - Fuel rails and injectors are over-pressurized.
  - Pressure regulator response time too slow.
  - Shutoff valve does not close completely.
  - *The FICM shall monitor fuel rail pressure* CM during CNG operation.
    - *Pressure regulator response time shall be a maximum of 25 ms.*
    - *The FICM shall monitor fuel tank pressure CM after shutoff valve is commanded <OFF>.*
- Inadequate process behavior (CNG engine):
  - The controlled process has other actuators from other systems affecting it.
  - The fuel injectors do not provide proper pressure.
  - The fuel rail has a connector leak that leads to loss of fuel containment.
    - *The FICM shall have prioritization logic to deal with multiple actuator actions.*
    - *The FICM shall monitor fuel rail pressure CM during CNG operation.*
    - *The FICM shall monitor fuel rail pressure CM during gasoline mode by use of fuel rail pressure sensors.*

Solutions for causal factors of loss scenarios can be treated as new requirements or constraints. New requirements or constraints derived by STPA are incorporated into the system design and are useful for updating the system architecture. The results of STPA record traceability as a rationale for the derived requirements.

Management of requirements traceability between those requirements generated by an STPA process and those requirements generated from other processes is an important activity. For example, requirement information may be managed by updating MBSE system models or reflecting it in a requirements management tool.

6.12 Lessons Learned

This section lists lessons learned by STPA practitioners that may be useful to new or other STPA practitioners:

- Leverage systems engineering principles.
  - Guidance (or examples) on creating control structure using systems engineering foundations.
- Begin with simple control structures and then go deeper.
  - Explain how to capture the system or function and create a control structure as a recommendation.
    - Provide examples of control structure, hierarchical layered structure versus non-layered structure. This may help STPA beginners to understand which control structure is more verifiable.
- Use STPA early in the engineering design process.
- Identify and engage a “chief system architect” for the system under review if available.
- Use collateral from existing safety process tasks to avoid repeating in STPA. (For example, some companies use PHA and HARA to identify potential accidents and hazards and to assess their associated risk levels. This info can be inserted into the STPA process to avoid parallel development of the same information.)

- Leverage STPA outputs in verification and validation test plans and test cases.
- STPA can complement, rather than replace, other methods—independent check, looking at safety problems from multiple viewpoints.
    - For example, HAZOP may identify hazards based on function, while STPA may identify hazards based on unsafe control actions. These hazard identification techniques can be performed independently. Comparing the results can (1) make sure neither method overlooked a hazard, and (2) help develop more rigorous definitions of hazards (i.e., what does the hazard cover/not cover).
- Incorporation of safety requirements into appropriate technical documents for system content design, supplier-provided entities, and validation test cases and planning.
- Visualize the loss scenarios.
    - Even if the UCA occurrence scenarios are complex due to interactive interference, it is possible to explain them systematically in an easily understandable way by summarizing them from the control flow viewpoint. It is effective as a safety argument.
    - The visualized loss scenario is helpful for planning the verification and validation, such as preparing the testing environments and generating test Cases.

6.13 Questions to Prepare for STPA

This section lists questions that may be useful to STPA practitioners to prepare for an STPA evaluation.

- What is the goal or mission of the system?
- What does the system look like from an architecture, functional, process, or controls structure perspective?
- Is there an operational description of the system's expected functions, behaviors, capabilities, and interactions with other systems, users, and human operators over expected operating scenarios?
- What are the required and expected functions for each of the system elements?
- Is there a priority of command to be maintained between different sub-systems during normal operation? An example would be holding a vehicle stationary on a hill with brakes and then wanting to accelerate up the hill; when are brake commands prioritized above accelerator pedal commands?
- What accidents is the system capable of causing?
- What are the hazardous conditions that may lead to any of these accidents?

6.14 Questions While Performing STPA

This section lists questions that may be useful to STPA practitioners to perform an STPA evaluation.

- What happens to the system when a system element fails or misbehaves?
- Does the resulting system behavior create a potentially hazardous condition(s)?
- What are the risks and risk assessments associated with these potential hazards?
- Which part of the system(s) becomes responsible for the failed/misbehaving element's functions?
- What are the resulting system capabilities after the failure/misbehavior?
- Do these resulting system capabilities create new/other potentially hazardous situations?

- Are these resulting system capabilities sufficient for achieving system goals/missions?
- What are the required diagnostic strategies and level of rigor necessary to detect failures or misbehaviors?
- What mitigation strategies and actions are necessary to transition the system from a detected failed/misbehaving state to a known, acceptable safe state?
- Based on the severity and exposure factors of the risk assessment (and potential operator controllability where applicable), is the residual risk acceptable?

6.15 Lessons Learned for Incorporating STPA in Large Organizations

This section outlines some of the experiences certain STPA practitioners associated with the development of this document have had when trying to incorporate STPA into large organizations. The following items can be helpful when dealing with this effort (Vernacchia, 2019). (Used with permission from Vernacchia. All other rights reserved.)

- First and foremost, make sure there is a need STPA can fill (e.g., HMI - socio-technical benefits, etc.).
- Don't try to change the whole world. The goal should be not to solve world hunger but just to feed the family.
- Maintain your vision, but be ready to modify based on good feedback or input.
- Leverage the idea of continuous improvement for existing processes by enhancing the use of systems engineering and systems thinking.
- Talk to other people inside and outside of your organization.
- Emphasize that STPA is a systematic approach to evaluate proposed system content for that content's ability to detect and mitigate a situation that could lead to potentially hazardous conditions.
- Propose STPA as an alternative to struggling, limited, or constrainted methodology.
    - For example, use STPA as an alternative to DFMEA's effort to deal with human factors.
- Operate below the "radar."
    - Be focused.
    - Do not alienate people with grandiose statements.
    - Be respectful of people's concerns.
- Look at other company successes (Boeing, Embraer, Ford, etc.).
- Seek out like-minded STPA practitioners in your industry or across industries to find common interests and needs.
    - SAE STPA Recommended Practices Task Force.
- Demonstrate the value of STPA requirements addressing safety concerns.
- Associate STPA with corporate initiatives when it helps those initiatives.
- Use on programs with new functions and features that have not been implemented yet or implemented together.

- Gather objective data showing results.
    - Requirements generated.
    - STPA evaluations drive design updates and changes ns.
    - Short time to get results.

6.16 Japan Automotive Software Platform and Architecture (JASPAR) Lessons Learned

The STPA Task Force coordinated its efforts with other organizations using STPA in their functional safety processes. The following content is feedback from one such group, JASPAR, that has a Functional Safety workgroup whose purpose is to "improve functional safety development in terms of quality, workload, and speed in order to drive functional safety for enhancing vehicle system."

It is effective to define a general-purpose control structure to support the STPA introduction. Prepare generic control structures with different types of hierarchies and use them according to the purpose of analysis. For distributed development in complex and large-scale systems, the granularity of analysis is unified to promote common understanding and to communicate with stakeholders.

Examples of general-purpose STPA control structures in the automotive domain. The features are as described below.

1. Hierarchical control structure.

   Established a hierarchical control structure where the vehicle level is ranked higher, and the system level is ranked lower, with the two levels being interrelated.

2. Vehicle level control structures that incorporate comprehensive viewpoints.

   Components consist of driver/systems/environments/vehicles that should be considered a starting point for large-scale systems.

3. System-level control structures.

   The components are classified into different legends to clarify which elements relate to one's company and which do not.

Example 1: If one's company's products are not relevant to the environment, descriptions of the environment are unnecessary.

Example 2: If there are interactions with ECUs made by other companies, use the component names of the other ECUs.

Example 3: Solid lines indicate interactions up to HMI, which relates to one's company's products, and dotted lines indicate the subsequent interactions between HMI and people.

6.16.1 Control Structure Legend

Table 2 - JASPAR lessons learned control structure legend

No	Shape	Explanation	Case
1		Human Controller	Driver, crew, dealer mechanic
2		System Controller	System, ECU, sensor, switch
3		Controlled process	Vehicle, mechanical parts, tires, brake pads, gearbox
4		Environment	Environmental factors (roads, pedestrians, weather, rear seat crews)
5		Interaction	Interaction
6		Related System	Handle as a black box (systems made by other companies, etc.)
7		Related Interaction	Assumed interactions (interactions involving systems made by other companies)
8		Contents of interaction	Driver operation on/off

6.16.2 Vehicle Level Control Structure

The important thing is to start STPA from a layer with a high level of abstraction regardless of the system scale. In other words, designating drivers, systems, vehicles, and environments as components adds a comprehensive field of vision to the analysis. It is one of the general-purpose control structure features.

The relationship between a vehicle's controller and controlled processes is expressed as a structure. Everything except for special systems such as airbags can be expressed using this general-purpose control structure. The relationship between the controller and the controlled processes is established between the driver and the systems, and a relationship is also established between the systems and the vehicle.

The environment is associated with the driver and system inputs and vehicle outputs.

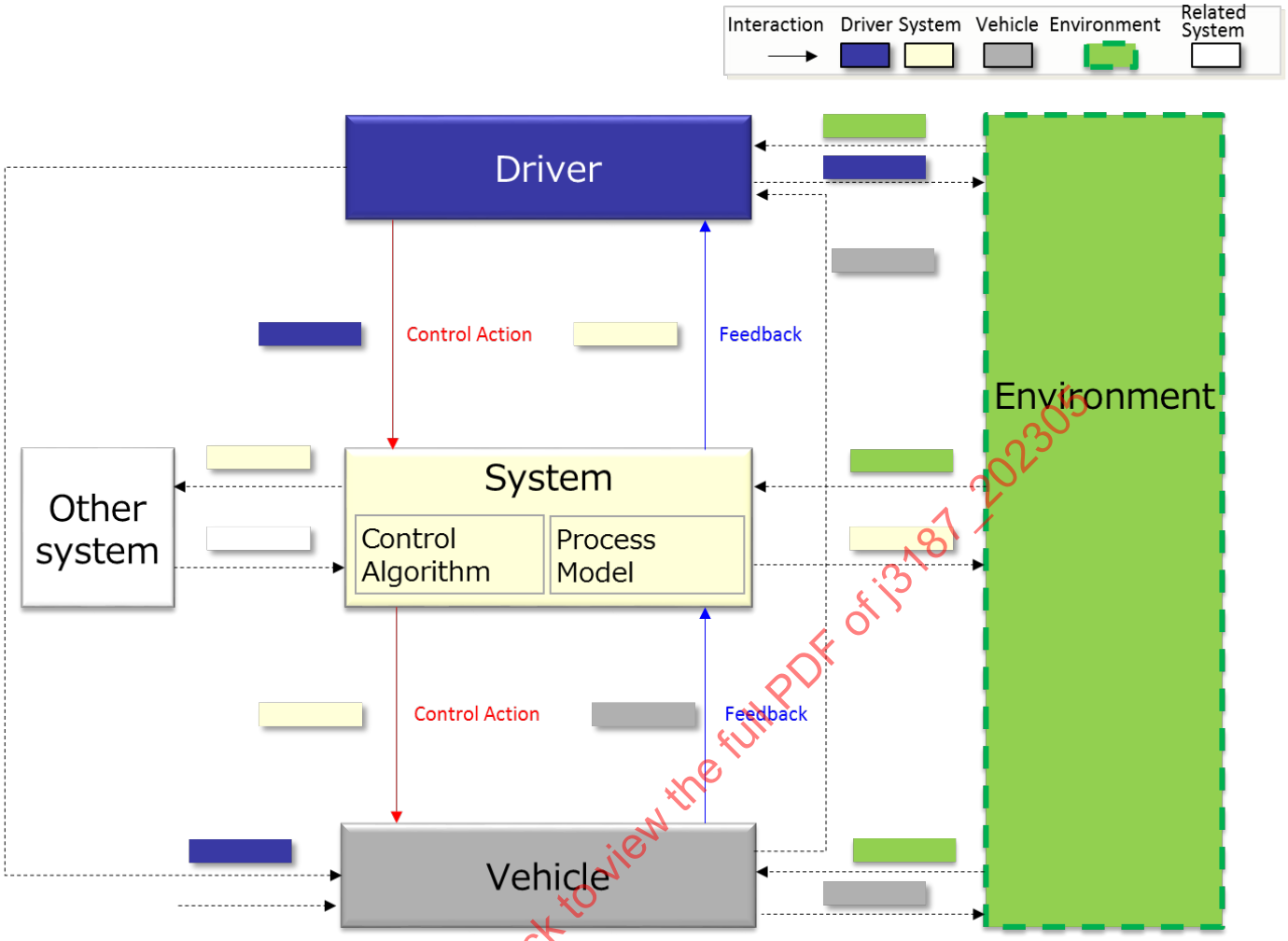

Figure 16 - JASPAR lessons learned vehicle level control structure

6.16.3 System-Level Control Structure

The system's relationship between the controller and controlled processes is also expressed as a structure. It is necessary to describe the interactions even for components with no direct association with the relevant systems, these are expressed by changing the color of the system components. (See Table 2.) The environment is associated with the driver and system inputs and vehicle outputs. The environmental components may be omitted if the relevant system has no association with the environment.

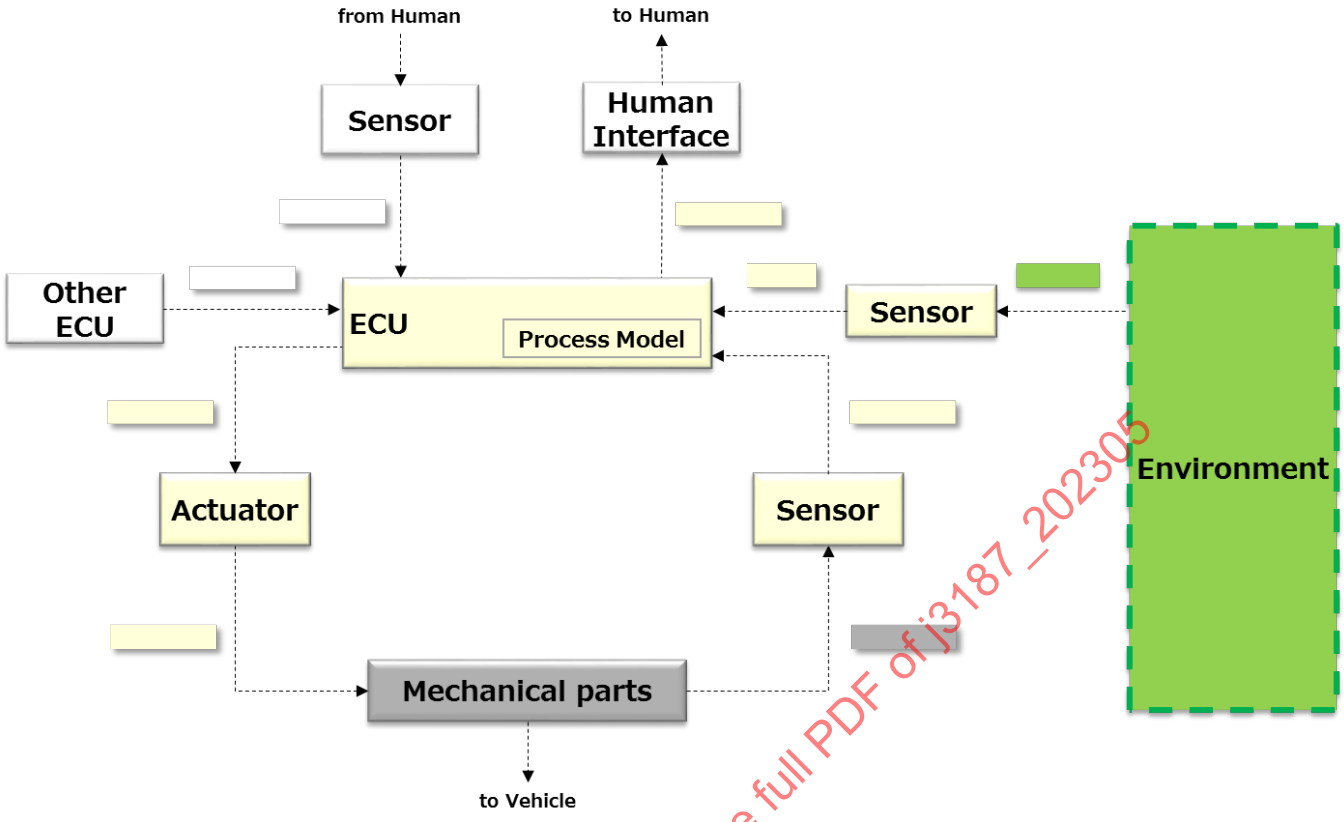

Figure 17 - JASPAR lessons learned system-level control structure

When MBSE is adopted for system design, system models such as preliminary architecture can be referred to the control structure development.

Components of the control structure can be easily extracted from the system model so that a control structure can be constructed more efficiently.

The system's behavior (e.g., state machine diagrams, sequence diagrams, and activity diagrams of omg sysml) helps to make communication smooth with reviewers and provides a foundation for the control structures and interactions of the analysis target systems.

7. HIGH-LEVEL USE OF STPA WITHIN SAFETY PROCESSES AND STPA WITH OTHER SAFETY EVALUATION METHODS

7.1 Introduction

This section describes several topics regarding how STPA could be used within a system safety process. Comparisons to other system safety analysis methodologies are highlighted, and STPA practitioners' experience regarding using STPA within safety processes is provided. Also illustrated are examples of requirement definition activities in various industry system safety evaluation standards and how STPA could be used to help those activities.

In addition, this section illustrates how STPA practitioners have found STPA to compare to other system safety evaluation methods and outlines various methods for using STPA and these other evaluation methods as part of a system safety analysis effort.

These topics have been the subject of much discussion over the past few years as more practitioners have had the chance to apply STPA to more applications. Chapter 3 of the STPA Handbook discusses the integration of STPA with a systems engineering process and provides the following useful illustration on page 54 of the STPA Handbook. This perspective represents an integration of STPA throughout the whole engineering process, illustrating that STPA can be integrated into all parts of an engineering process.

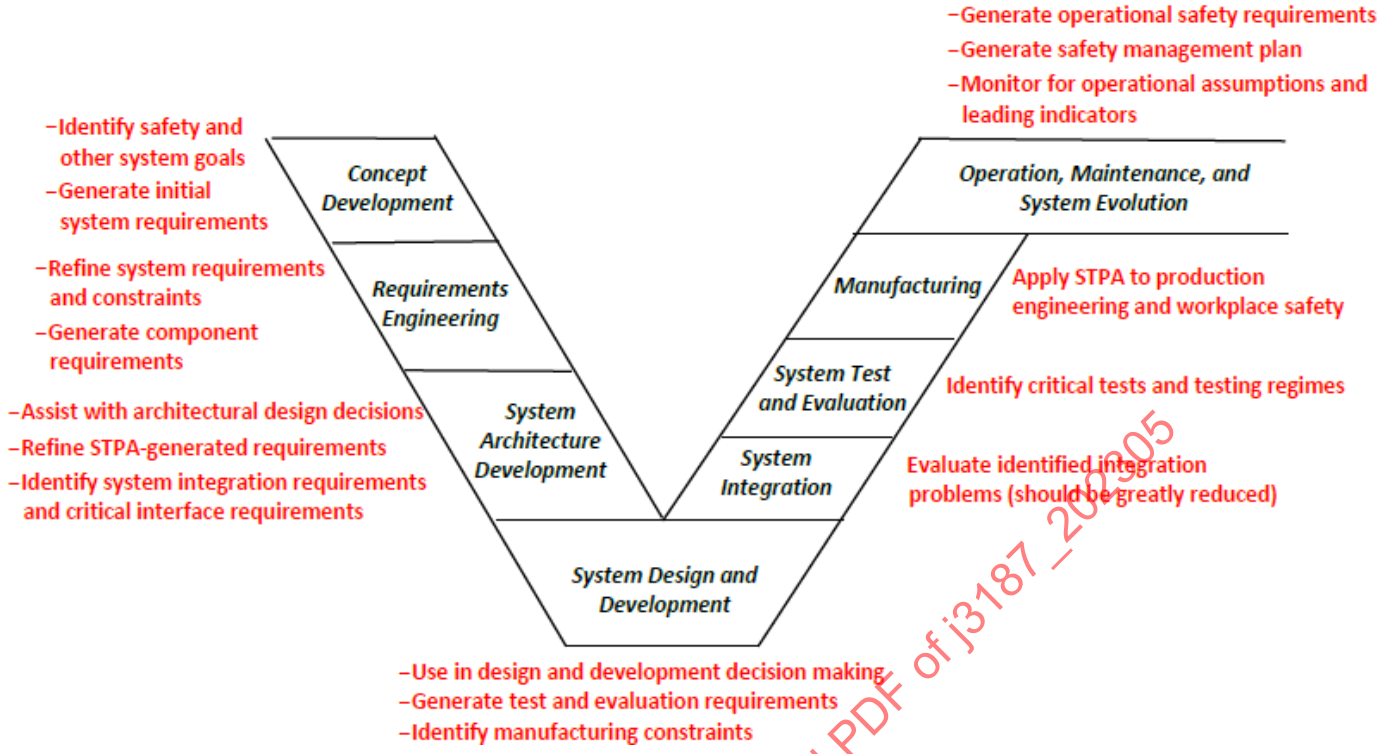

Figure 18 - Systems engineering "V" model and STPA (Leveson and Thomas, 2018)
Used with permission from Leveson and Thomas. All other rights reserved.

Chapter 3 in the STPA Handbook goes on to provide the following guidance shown in *italics* here:

> *STPA can be used throughout the standard system engineering process, starting in the earliest concept development stage. In essence, STPA can be used to generate high-level safety requirements early in the concept development phase, refine them in the system requirements development phase. The system requirements and constraints can then assist in the design of the system architecture and more detailed system design and development. The STPA result process then goes hand in hand with design and development as the analysis can be used to inform decisions as they are made. STPA continues to be useful through assurance and manufacturing and provides important information for use during operations.*
>
> *STPA fits into a model-based engineering process as it works on a model of the system (which is refined as design decisions are made), although that model is different than the architectural models usually proposed for model-based system engineering today. STPA promotes traceability throughout the development process so decisions and designs can be changed with minimum requirements for redoing previous analyses.*
>
> *Finally, as noted in many places in this handbook, STPA can be applied to any emergent system property in the system engineering and product lifecycle, not just safety.*
>
> *STAMP and STPA contribute to all the activities in system engineering. The use of the analysis and its results for the following activities are described in the rest of this chapter:*
>
> 1. *Definition of losses to be handled during development and operation*
> 2. *Identification of external constraints (including market and regulatory requirements) on the system design*
> 3. *Identification of system-level hazards and the related requirements and constraints on system behavior*
> 4. *Modeling of the system control structure*
> 5. *Refinement of hazards and constraints and allocation of functions to system components*

6. *Assist with making architectural, design, and implementation decisions*
7. *System Integration assistance*
8. *Generation of system test requirements*
9. *Control of manufacturing (production engineering, workplace safety)*
10. *Generation of operational safety requirements (including leading indicators of increasing risk) and the safety management plan*
11. *Operational safety management, including monitoring leading indicators*

Used with permission from Leveson and Thomas. All other rights reserved.

However, full integration of STPA into an engineering process may encounter organizational or societal constraints and resistance within some organizations. Many organizations feel they have well-established processes that would not benefit from integrating STPA into these processes. In addition, some organizations have system evaluation methodologies deeply integrated into their engineering processes and perceive the addition of STPA to disrupt such processes. See 6.15 for more insight regarding these situations.

7.1.1 STPA Assistance in Established System Safety Processes and/or Standards

Various system safety processes and/or standards have been known to the safety community for several years. ISO 26262 is familiar to automotive system safety engineers in the automotive area. The ISO 26262 series of standards is the adaptation of the IEC 61508 series of standards to address the sector-specific needs of electrical and/or electronic (E/E) systems within road vehicles. This adaptation applies to all activities during the safety lifecycle of safety-related systems comprised of electrical, electronic, and software components.

ISO 21448:2022 is a recent ISO standard targeted at potentially hazardous behavior caused by the intended functionality or performance limitation of a system that is free from faults that would be addressed by ISO 26262 content. ISO 21448:2022 cites: (1) the inability of the function to correctly comprehend the situation and operate safely, including functions that use machine learning algorithms, and (2) insufficient robustness of the function with respect to sensor input variations or diverse environmental conditions as examples of intended functionality or performance limitations. While ISO 21448:2022 considers HMI in the case of incorrect usage and misusage, STPA expands this to consider also further risks such as accidental operations.

MIL-STD-882E is another familiar document covering the Department of Defense (DoD) systems engineering (SE) approach to eliminating hazards where possible and minimizing risks where those hazards cannot be eliminated.

ARP4761 is also a familiar document to the aviation community. ARP4761 mainly focusses on the safety methods. The overall design process that integrates safety is provided by ARP4754.

These standards have some basic common objectives where initial system information, objectives, and stakeholder requirements are gathered and evaluated; potential hazards are identified, hazards are assessed, and risk levels are defined, requirements are developed to prevent or manage these hazards to an acceptable level, test cases are developed to verify compliance to requirements and methods to document objective data to support risk management suggested. Table 3 shows a high-level summary of ISO 26262, MIL-STD-882E, and ARP4761 activities related to requirements development and definition. Comparisons to ISO 21448:2022 will be included in the next version of this document.

It should be noted that Table 3 represents a high-level summary of ISO 26262, MIL-STD-882E, and ARP4761A requirements development activities and is provided for comparison purposes only. Table 3 is not intended to portray the complete content of any of the standards. More information regarding each standard may be found by consulting a copy of the full standard.

NOTE: ARP4761 describes guidelines and methods of performing the safety assessment for certification of civil aircraft. For reference, ARP4754 discusses the development of aircraft systems considering the overall aircraft operating environment and functions.

Table 3 - Standards comparison for requirements definition

ISO26262	MIL-STD-882E	ARP4761A
Part 3 - 5.4.1 - System Information	**System Information**	**System Information**
"Item" ("System" or "Combination of Systems") Requirements Made Available	ConOPs Review	Descriptions of Aircraft Functions Derived from the Aircraft Design Objectives
Legal, National , International Standards	Stakeholder Expectations	Descriptions of System Functions decomposed from the aircraft functions
Functional Behavior at Vehicle level		Fundamental Necessities of Flight Due to Laws of Physics
Required Quality, Performance, Availability		Operational and Environmental Conditions Aircraft is Designed to Encounter
Functional Dependencies		Description of aircraft architecture (or proposal)
Operating Environments		Description of systems architectures (or proposals)
Known Failure Modes and Hazards		Initial crew procedures
Part 3 - 5.4.2 "Item" Boundary, Interfaces, Interactions		
Effects of Behavior on Vehicle		
Functionality Required by Other Items		
Functional of Other Items Required by Item		
Functional Dependencies		
Operating Allocation of Functions Amongst Involved Systems and Elements		
Operational Scenarios Which May Impact Item Functionality		
Part 3 - 6.1 - Hazard Analysis and Risk Assessment	**Task 201 – Preliminary Hazard List**	**Aircraft Functional Hazard Assessment (AFHA)**
Objectives of This Clause	Hazard Identification	Identifies the Failure Conditions Associated with Each Aircraft Level Function over its operational phases
Identify and Classify Hazardous Events Caused by Item Malfunctioning Behavior	Brief Description of Hazard	Classifies the Failure Conditions Associated with Each Aircraft Level Function
Formulate Safety Goals, with Corresponding ASILs, Related Prevention or Mitigation of Hazardous Events to Avoid Unreasonable Risk	Causal Factors for Each Hazard	Formulate Classification Establishes Safety Objectives Aircraft Must Meet Classification is Severity Based
		Captures expected crew responses in establishing effects and severities
		Identifies and Evaluates Potential Hazards Related to Aircraft to establish safety objectives to evaluate its Design

ISO26262	MIL-STD-882E	ARP4761A
Part 3 - 6.2 - Hazard Analysis and Risk Assessment	**Task 202 - Preliminary Hazard Analysis**	**Preliminary Aircraft Safety Assessment (PASA)**
Evaluate Without Internal Safety Mechanisms Situational Analysis	Initial Risk Assessment Severity	Determine How Failures Can Lead To Aircraft Level Failure Conditions Identified by the AFHA
Correct Usage	Probabilities	Assesses How Their Failures Can Lead to Aircraft Level Failure Conditions Identified by the AFHA
Reasonable Incorrect Usage	Risk Assessment Codes (RACs)	Identifies the Interactions And Dependencies Between the Aircraft Systems
Systematic Hazard Determination	Risk Mitigation Using Design Order of Precedence	Determine whether the Proposed aircraft Architecture Can Reasonably Be Expected To Meet Safety Objectives Identified by AFHA - uses analysis methods such as shallow Numerical Probability Analyses, high-level FMEAs, Particular Risk Analyses and Common Mode Analysis.
Based on Possible Item Malfunctions		Establishes the Safety Requirements of the Aircraft for architectural features and probability allocations needed for architecture to meet those objectives.
Methods (e.g., HAZOP, FMEA, etc.)		Assigns Functional Development Assurance Levels (FDALs) for aircraft level functions.
Hazards Identified at Vehicle Level Hazardous Event Consequences Identified Risk Classification (ASIL) Severity (Human Harm) Exposure (Operational Likelihood) Controllability (Driver Influence) Safety Goals Development Prevent Hazard Occurrence Mitigate Hazards to Avoid Unreasonable Risk Safety Goals Represent Top-Level Safety Req'mts		
Part 3 - 7.1 - Functional Safety Concept - Objectives	**Task 203 - System Req'mt Hazard Analysis**	**System Functional Hazard Assessment (SFHA)**
Specify Functional or Degraded Functional Behavior in Accordance with Item's Safety Goals	Determine System Design Requirements	Identify Failure Conditions of each system level function over the aircraft's operational phases, including those fulfilling any aircraft level functions allocated to the system
Specify Constraints and Requirements for Suitable and Timely Detection and Control of Relevant Faults in Accordance with Safety Goals	Eliminate Hazard Potential	Classification of these Failure Conditions are severity based and establishes safety objectives system must meet
Allocate Functional Safety Requirements to System Architectural Design	Reduce Associated Risks	Identifies and Evaluates Potential Hazards Related to Aircraft to establish safety objectives to evaluate Its Design
	Incorporate Approved Design Requirements into Appropriate Design Documentation Address Requirements in all Appropriate Technical Reviews	Captures expected crew responses in establishing effects and severities

ISO26262	MIL-STD-882E	ARP4761A
Part 4 - 6 - Technical Safety Concept - Objectives	**Task 205 - System Hazard Analysis**	**Preliminary System Safety Assessment (PSSA)**
Specify technical safety requirements regarding the functionality, dependencies, constraints and properties of system elements & interfaces needed for implementation	Verify System Compliance with Requirements to Eliminate or Reduce the Associated Risks	Determine how failures can lead to system level Failure Conditions identified by the SFHA
Specify technical safety requirements regarding the safety mechanisms to be implemented in system elements and interfaces	Identify Previous Unidentified Hazards Associated with Sub-System Interfaces and Faults	Determine whether the proposed architecture can reasonably be expected to meet the safety objectives identified by SFHA and safety requirements allocated from the PASA - uses analysis methods such as shallow Numerical Probability Analyses, high-level FMEAs, Particular Risk Analyses and Common Mode Analysis. Scope includes both the applicable system and the interfacing resources it uses (e.g. power, networks, sensors, displays)
Specify requirements regarding the functional safety of system and its elements during production, operation, service and decommissioning	Identify Hazards Associated with Integrated System Design Including Software and Sub-System Interfaces	Establishes the Safety Requirements of the system for architectural features and probability allocations needed for its architecture to meet those objectives.
Verify that the technical safety requirements are suitable to achieve functional safety at the system level and are consistent with the functional safety requirements	Recommend Actions Necessary to Eliminate Identified Hazards or Mitigate Associated Risks	Assigns Functional Development Assurance Levels (FDALs) for system level functions and Item Development Assurance Levels (IDALs) for items within the system that use RTCA DO-178C or RTCA DO-254 for their assurance.
Develop a system architectural design and a technical safety concept that satisfy the safety requirements and that are not in conflict with non-safety-related req'mts	**Task 208 - Functional Hazard Analysis**	
Analyze the system architectural design in order to prevent faults and to derive the necessary safety-related special characteristics for production and service	Decomposition of System and Related Sub-Systems to Major Component Level	
Verify that the system architectural design and the technical safety concept are suitable to satisfy the safety requirements according to respective ASIL	Functional Description of Each Sub-System and Component	
	Functional Description of Interfaces Between Sub-Systems and Components	
	Hazards evaluated for loss of function, degraded function or malfunction, or functioning out of time or out of sequence for the subsystems, components, and interface	
	Assessment of Risk Associated with Each Identified Failure of a Function, Sub-System, or Component	
	List of Requirements and Constraints that, when Successfully Implemented, Will Eliminate Hazard or Reduce Risk	
Human Machine Interactions	**Human Machine Interactions**	**Human Machine Interactions**
No Such Specific Section	**No Such Specific Section**	Crew responses are captured in the AFHA and SFHA *as needed* to establish failure condition classifications so that the PASA and PSSA can identify necessary crew procedures and system requirements for flight deck effects and human interfaces needed for the crew to perform those actions.

Table 4 illustrates the key steps of an STPA evaluation. It also highlights that STPA enables HMI evaluations and for the evaluation of system scenarios where nothing has failed, something not covered in ISO 26262. As described in 8.8.2, ISO 21448:2022 attempts to fill this gap. Still, STPA expands this to consider further risks as accidental operations. Also, ISO 21448:2022 is intended for complex systems and processing algorithms, but not all E/E systems and components fall within that scope. Therefore, ISO 21448:2022 is not applied to all products to that ISO 26262 is applied.

Table 4 - STPA high-level main steps

STPA
System Information
Define System Scope & Boundary
System to Sub-Systems Relationships
Expected Systems Functions
Define Purpose
Identify Losses
Identify System Level Hazards
Identify System Safety Constraint
Refine Hazards if Needed
Modeling Control Structure
Control Hierarchy
Control Actions
Commands (Flow Down)
Feedback (Flow Up)
Controller Content
Control Algorithm(s)
Process Models
Controlled Process(es)

STPA
Identify Unsafe Control Actions (UCAs)
System Elements Functional Definitions
Use of Guideword Analysis
What Leads to Hazard(s)?
Constraints and Requirements to Prevent or Mitigate Hazard
Identify Loss Scenarios
Why Would UCAs Occur?
Failure Related to Controllers
Inadequate Control Algorithm(s)
Unsafe Inputs from Other Controllers
Inadequate Process Model(s)
Why Would UCAs Be Improperly Executed or Not Executed?
Control Action Not Executed by Actuators
Control Action Improperly Executed
Unsafe Inputs from Other Controllers
Controlled Process Does Not Respond
Controlled Process Responds Improperly
Constraints and Requirements to Prevent or Manage Scenarios That May Lead to Hazards

STPA
Human Machine Interactions
Representation of Human as Control System Element
Human Extension Representation
Control Action Selection
Mental Models
Process State
Process Behavior
Environment
Mental Model Updates

Table 5 juxtaposes the steps shown in Table 4 to the standard's activities in Table 3. Table 5 illustrates that steps are synergistic to the standard activities in that both STPA and the standards focus on developing requirements to prevent or manage hazards to an acceptable level. It is not intended to convey that STPA is a safety process such as the standards included, but rather that STPA activities as an analysis methodology are synergistic to activities outlined in these standards.

Table 5 - STPA with ISO 26262, MIL-STD-882E, and ARP4761

STPA	ISO26262	MIL-STD-882E	ARP4761A
System Information	**Part 3 - 5.4.1 - System Information**	**System Information**	**System Information**
Define System Scope & Boundary	"Item" ("System" or "Combination of Systems") Requirements Made Available	ConOPs Review	Descriptions of Aircraft Functions Derived from the Aircraft Design Objectives
System to Sub-Systems Relationships	Legal, National , International Standards	Stakeholder Expectations	Descriptions of System Functions decomposed from the aircraft functions
Expected Systems Functions	Functional Behavior at Vehicle level		Fundamental Necessities of Flight Due to Laws of Physics
	Required Quality, Performance, Availability		Operational and Environmental Conditions Aircraft is Designed to Encounter
	Functional Dependencies		Description of aircraft architecture (or proposal)
	Operating Environments		Description of systems architectures (or proposals)
	Known Failure Modes and Hazards		Initial crew procedures
	Part 3 - 5.4.2 "Item" Boundary, Interfaces, Interactions		
	Effects of Behavior on Vehicle		
	Functionality Required by Other Items		
	Functional of Other Items Required by Item		
	Functional Dependencies		
	Operating Allocation of Functions Amongst Involved Systems and Elements		
	Operational Scenarios Which May Impact Item Functionality		
Define Purpose:	**Part 3 - 6.1 - Hazard Analysis and Risk Assessment**	**Task 201 – Preliminary Hazard List**	**Aircraft Functional Hazard Assessment (AFHA)**
Identify Losses	Objectives of This Clause	Hazard Identification	Identifies the Failure Conditions Associated with Each Aircraft Level Function over its operational phases
Identify System Level Hazards	Identify and Classify Hazardous Events Caused by Item Malfunctioning Behavior	Brief Description of Hazard	Classifies the Failure Conditions Associated with Each Aircraft Level Function
Identify System Safety Constraint Refine Hazards if Needed	Formulate Safety Goals, with Corresponding ASILs, Related Prevention or Mitigation of Hazardous Events to Avoid Unreasonable Risk	Causal Factors for Each Hazard	Formulate Classification Establishes Safety Objectives Aircraft Must Meet Classification is Severity Based
			Captures expected crew responses in establishing effects and severities
			Identifies and Evaluates Potential Hazards Related to Aircraft to establish safety objectives to evaluate its Design

STPA	ISO26262	MIL-STD-882E	ARP4761A
Modeling Control Structure	**Part 3 - 6.2 - Hazard Analysis and Risk Assessment**	**Task 202 - Preliminary Hazard Analysis**	**Preliminary Aircraft Safety Assessment (PASA)**
Control Hierarchy Control Actions Commands (Feed Down) Feedback (Feed Up) Controller Content Control Algorithm(s) Process Models Controlled Process(es)	Evaluate Without Internal Safety Mechanisms Situational Analysis Correct Usage Reasonable Incorrect Usage Systematic Hazard Determination Based on Possible Item Malfunctions Methods (e.g., HAZOP, FMEA, etc.) Hazards Identified at Vehicle Level Hazardous Event Consequences Identified Risk Classification (ASIL) Severity (Human Harm) Exposure (Operational Likelihood) Controllability (Driver Influence) Safety Goals Development Prevent Hazard Occurrence Mitigate Hazards to Avoid Unreasonable Risk Safety Goals Represent Top-Level Safety Req'mts	Initial Risk Assessment Severity Probabilities Risk Assessment Codes (RACs) Risk Mitigation Using Design Order of Precedence	Determine How Failures Can Lead To Aircraft Level Failure Conditions Identified by the AFHA Assesses How Their Failures Can Lead to Aircraft Level Failure Conditions Identified by the AFHA Identifies the Interactions And Dependencies Between the Aircraft Systems Determine whether the Proposed aircraft Architecture Can Reasonably Be Expected To Meet Safety Objectives Identified by AFHA - uses analysis methods such as shallow Numerical Probability Analyses, high-level FMEAs, Particular Risk Analyses and Common Mode Analysis. Establishes the Safety Requirements of the Aircraft for architectural features and probability allocations needed for architecture to meet those objectives. Assigns Functional Development Assurance Levels (FDALs) for aircraft level functions.
Identify Unsafe Control Actions (UCAs)	**Part 3 - 7.1 - Functional Safety Concept - Objectives**	**Task 203 - System Req'mt Hazard Analysis**	**System Functional Hazard Assessment (SFHA)**
System Elements Functional Definitions Use of Guideword Analysis What Leads to Hazard(s)? Constraints and Requirements to Prevent or Mitigate Hazard	Specify Functional or Degraded Functional Behavior in Accordance with Item's Safety Goals Specify Constraints and Requirements for Suitable and Timely Detection and Control of Relevant Faults in Accordance with Safety Goals Allocate Functional Safety Requirements to System Architectural Design	Determine System Design Requirements Eliminate Hazard Potential Reduce Associated Risks Incorporate Approved Design Requirements into Appropriate Design Address Requirements in all Appropriate Technical Reviews	Identify Failure Conditions of each system level function over the aircraft's operational phases, including those fulfilling any aircraft level functions Classification of these Failure Conditions are severity based and establishes safety objectives system must meet Identifies and Evaluates Potential Hazards Related to Aircraft to establish safety objectives to evaluate Its Design Captures expected crew responses in establishing effects and severities