SYSTEMS-THEORETIC PROCESS ANALYSIS AND SAFETY-GUIDED DESIGN OF MILITARY SYSTEMS

by

David Craig Horney

B.S. Aeronautical Engineering
United States Air Force Academy, 2015

SUBMITTED TO THE DEPARTMENT OF AERONUATICS AND ASTRONAUTICS IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE
AT THE
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
June 2017

Signature of Author: ______________________________
Department of Aeronautics and Astronautics
May 1, 2017

Certified by: ______________________________
Nancy Leveson
Professor of Aeronautics and Astronautics and Engineering Systems
Thesis Supervisor

Accepted by: ______________________________
Youssef M. Marzouk
Associate Professor of Aeronautics and Astronautics
Chair, Graduate Program Committee

Page intentionally left blank.

SYSTEMS-THEORETIC PROCESS ANALYSIS AND SAFETY-GUIDED DESIGN OF MILITARY SYSTEMS

by
David Craig Horney

B.S. Aeronautical Engineering
United States Air Force Academy, 2015

Submitted to the Department of Aeronautics and Astronautics on May 1, 2017
in Partial Fulfillment of the Requirements for the
Degree of Master of Science in Aeronautics and Astronautics

ABSTRACT:

Increasingly complex software enabled systems demand a new hazard analysis and safety-guided design technique in order to meet stringent safety standards and expectations. System Theoretic Process Analysis (STPA) proves to be a powerful tool to identify, describe and help mitigate hazards from the earliest conceptual development through the operations of a system. A future military aircraft example demonstrates STPA's applicability for preliminary hazard analysis, analysis of alternatives, organizational design, developmental test, and into operations. STPA is a hazard analysis framework that helps manage risks and safety responsibilities throughout the entire lifecycle of a system.

Thesis Supervisor: Nancy Leveson
Title: Professor of Aeronautics and Astronautics and Engineering Systems

Page intentionally left blank.

ACKNOWLEDGEMENTS

There are many people that I would like to thank who have helped me so much during my time here at the Massachusetts Institute of Technology (MIT). Thank you to Dr. John Thomas, Dr. Kip Johnson, Dr. Bill Young, Blake Abrecht, Andrea Scarinci, Diogo Castillo, Meg France, Sarah Folse and everyone else in the Systems Engineering Research Lab who has made the lab a welcoming and enjoyable place to conduct research. Thank you also to the entire military cohort here in Boston. Your friendships have made the time here at MIT fly by.

I would like to thank Alex Boydston and his team with the U.S. Army for helping provide subject matter expertise and support throughout this effort. Included on that team, a special thanks to Holt Busbee for helping arrange interviews with subject matter experts and Glenn Carter for his thorough reviews and comments. Thank you also to John Kuconis, Paula Ward and everyone else at MIT Lincoln Laboratory who made my time out here possible through the MIT Lincoln Laboratory Military Fellowship Program.

I am incredibly thankful for all of the support provided by my parents, Ed and Jacque, and my sister Allison throughout my time at MIT. They encouraged me to apply and have supported me in every step during my time in graduate school.

To my lovely wife Rebecca: marrying you here in Boston has been the biggest blessing of my life. I cannot imagine my life without you and am so excited that the rest of it includes you! Thank you for your constant love and support.

Finally, I owe a huge debt of gratitude to my advisor- Professor Nancy Leveson for allowing me the opportunity to attend MIT and for teaching me about systems engineering and safety. You enabled me fulfil my ambition to help improve safety in military aviation. Hopefully the fruits of our labor are felt for decades to come.

Page intentionally left blank.

Contents.

Table of Figures.

1. Introduction

The modern aircraft is a sight to behold; powerful engines, sleek sweeping lines, cockpits filled with incredible instrumentation and sensors. Along with this beauty and capability comes incredible complexity. The components interact in ways that are unexpected at first glance. It is impossible for one engineer to understand every minute detail of how a modern aircraft functions and eventually flies. This complexity leads to amazing capabilities but can also lead to hazardous and hard to predict scenarios. The analysis of complex systems requires a tool that can manage complexity while offering insight into the detailed operation of the system. One goal of a safety program is to reduce the vulnerability of the system while maintaining its capabilities. A method that puts safety at odds with capabilities will never be accepted.

This effort seeks to find ways to seamlessly incorporate safety into the design process for a new hypothetical aircraft. Robert Steiglitz wrote, “Safety must be designed and built into airplanes, just as are performance, stability, and structural integrity.” [1] The goal is to start the safety program at the very beginning of concept development and build a foundation of safety that will make the system more capable throughout its entire lifecycle. Achieving this goal not only reduces the likelihood of mishaps, but also reduces the cost of the program and help designers make well informed decisions.

This effort specifically investigates incorporating safety into the Department of Defense’s (DoD) engineering process, a well-defined and respected process that is used widely in industry as well. The aircraft of interest is a hypothetical light military transport aircraft contracted by the Army. It must be capable of carrying fourteen combat troops into battle in full gear. Each soldier will weigh approximately 350 pounds (lbs.) with their gear. The aircraft must also be capable of transporting a payload of 15,000 lbs. over a range of 800 nautical miles (nm) without outside support. It must be able to deliver troops and cargo to remote bases and land on unimproved runways with short take-off and landing (STOL) capability. All fourteen troops must be able to unload with their gear in 60 seconds. Lastly, the aircraft must be able to travel in a tethered formation. A single crew must be able to control three aircraft from takeoff to landing at improved airports with instrument landing system (ILS) capabilities.

The analysis presented fulfils tasks 201 and 202 in Military Standard (MIL-STD) 882E. These tasks are the earliest system safety tasks required in a project lifecycle. They call for hazards to be identified and for methods of mitigation to be investigated in the early stages of design. These tasks are traditionally fulfilled by making a Preliminary Hazard List (PHL) and by performing a Preliminary Hazard Analysis (PHA). In this effort Systems-Theoretic Process Analysis (STPA) will be used. STPA is a method based on systems theory that allows engineers to carefully and methodically analyze the hazards in a system. STPA has many benefits over the traditional safety analysis methods which will be discussed.

In addition to presenting the methodology for conducting an STPA analysis, this study demonstrates how to incorporate safety into the design process in a way that helps inform the design team of the safety implications of their decisions. This process allows safety to be considered in tradeoff analysis. The models produced during STPA analyses not only enhance safety, but improve the mental model of the system being designed. The STPA process creates a common model of the system for all members of the design team and highlights the feedback and control relationships that are necessary to design a well-functioning system. Incorporating safety into the design process using STPA is shown to enhance the design process in numerous ways, the most significant being the inclusion and integration of safety into all stages of design.

Lastly, this analysis demonstrates how STPA based methods can be applied throughout the lifecycle of the system in order to improve operations, guide developmental testing, and drive the safety management during operations. Summaries of relevant methods are provided and related back to the system of interest. Overall, this effort demonstrates how STPA can be of value during the system lifecycle and provides sources for detailed explanations where necessary.

2. Literature Review

The advanced complex machines that support the world we live in today demand rigorous systems engineering during their lifecycle. Systems engineering enables us to design machines that perform an incredible variety of tasks very well. Similarly, systems safety enables design teams to create machines that perform in dangerous environments and complete difficult missions while keeping the occupants from harm and protecting material assets. Our modern world is comprised almost entirely of systems that have associated benefits and risks. Every day we choose to accept certain risks and avoid others. In every system, engineers must weigh the benefits against the inherent risks. Systems safety engineering is comprised of processes that enable this risk assessment and help us make informed risk decisions. [2] As Clifton Ericson writes,

> "Risks are akin to the invisible radio signals that fill the air around us, in that some are loud and clear, some very faint, and some are distorted and unclear. Life, as well as safety, is a matter of knowing, understanding, and choosing the risk to accept. System safety is the formal process of identifying and controlling mishap risk. As systems become more complex and more hazardous, more effort is required to understand and manage system mishap risk." [2]

Because many of the risks around us are difficult to discern, it is incredibly important to develop and utilize powerful tools to find hazards in a system and eliminate or mitigate them through design.

Informally, systems safety has been practiced for centuries. Humans have always weighed the risks and rewards of their actions in order to make informed decisions. Systems safety was formalized as a discipline in the past sixty years as the machines we commonly utilize became more complex and potentially dangerous and simultaneously, our risk tolerance declined. [3] In the United States, the government and specifically the DoD has played a big role in shaping systems safety processes. MIL-STD-882 is the DoD standard for systems safety and has shaped safety processes across disciplines. Figure 1 illustrates the general systems safety process from MIL-STD-882E.

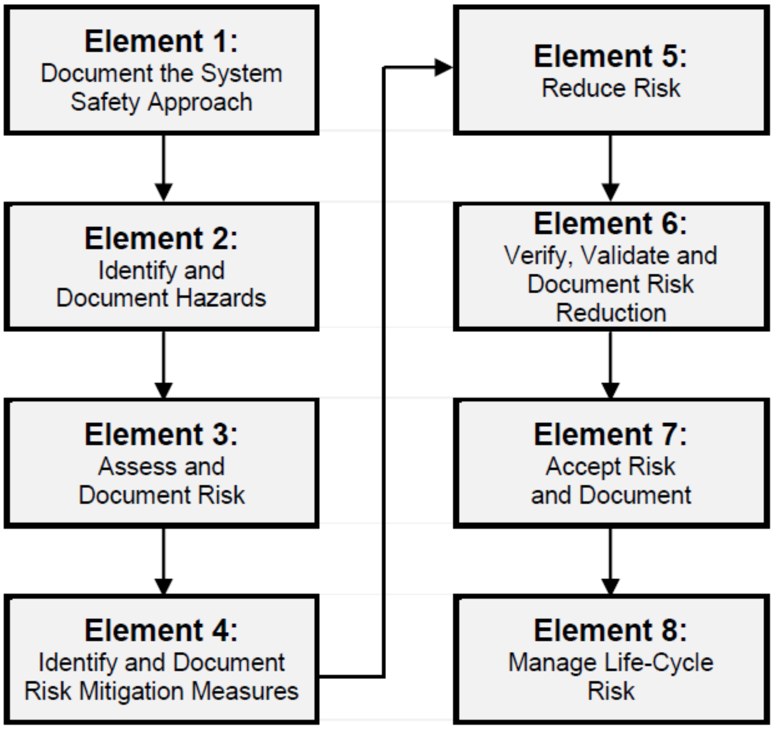


Figure 1. Eight elements of the system safety process. [4]

This effort focuses on systems safety during the early concept phase of the life cycle and will thus focus on elements one through three. MIL-STD-882E outlines tasks that must be completed but does not prescribe specific methods of analysis. The standard acts as a guide but leaves the specific analysis up to the program managers and systems engineers. This fact helps makes MIL-STD-882E a powerful and flexible standard that can be used for nearly any system.

MIL-STD-882E defines safety as “Freedom from conditions that can cause death, injury, occupational illness, damage to or loss of equipment or property, or damage to the environment.” [4] In order to achieve safe systems by avoiding mishaps, MIL-STD-882E prescribes tasks to be completed throughout the life of a project. The two most relevant analysis tasks during the conceptual stage of design are task 201, Preliminary Hazard List (PHL), and task 202, Preliminary Hazard Analysis (PHA). These tasks are completed early in the design of a program and set the stage for the rest of the systems safety analysis. These tasks are also the basis of the safety requirements for the project.

Many of the popular hazard analysis methods such as Failure Mode and Effects Criticality Analysis (FMECA) and Hazard and Operability analysis (HAZOP) are bottom-up approaches. This means that these techniques focus on low-level failures in the system and then trace faults to discover system level hazards and mishaps that may occur due to component failures. These methods are poorly suited for early design activities because of their need for design details for the system. Top down approaches are better suited for early concept analysis because they can start with top-level ideas and work towards more detailed hazards as the design matures. There are several top down approaches that have been developed and applied for systems safety.

Some common top down approaches rely on mature safety analyses or mature systems. Fault tree analyses (FTA) start with a hazard and use a tree like visualization of the system to determine what failures in the system could cause the hazards. This approach relies on a known list of hazards or mishaps as well as knowledge of how the system works and the components interact. This information is not available in the early stages of design. For this reason, these analysis tools are not commonly used until a system architecture has been established.

All of the listed analysis techniques are useful in the later stages of design to help drive reliability and safety in complex systems. Two commonly practiced hazard analysis techniques specifically fulfil tasks 201 and 202 in MIL-STD-882E. For decades, systems safety engineers have been using formalized techniques to build a PHL and perform a PHA on new systems. The techniques for these activities rely heavily on experienced engineers and input from subject matter experts. Figure 2 shows the general PHL process including inputs and outputs.

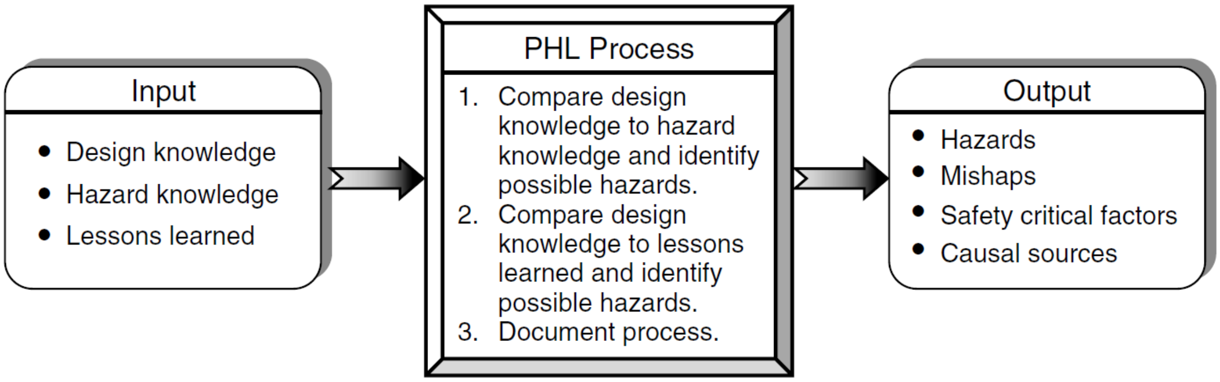

Input
• Design knowledge
• Hazard knowledge
• Lessons learned
PHL Process
1. Compare design knowledge to hazard knowledge and identify possible hazards.
2. Compare design knowledge to lessons learned and identify possible hazards.
3. Document process.
Output
• Hazards
• Mishaps
• Safety critical factors
• Causal sources

Figure 2. Preliminary hazard list overview. [2]

Most safety engineers use worksheets and organize brainstorming sessions with subject matter experts to develop preliminary hazard lists. Engineers use checklists of possible hazard sources to guide their discussion. An example checklist could include:

1. Energy sources
2. Hazardous functions
3. Hazardous operations
4. Hazardous components
5. Hazardous materials
6. Lessons learned from similar type systems
7. Undesired mishaps
8. Failure mode and failure state considerations [2]

These checklists along with worksheets and talented engineers allow for the identification of some of the hazards that are likely to arise during design and operations.

After a PHL has been completed and more detailed design is beginning, MIL-STD-882E calls for a PHA. PHAs rely on similar methods to PHLs but go into more detail.

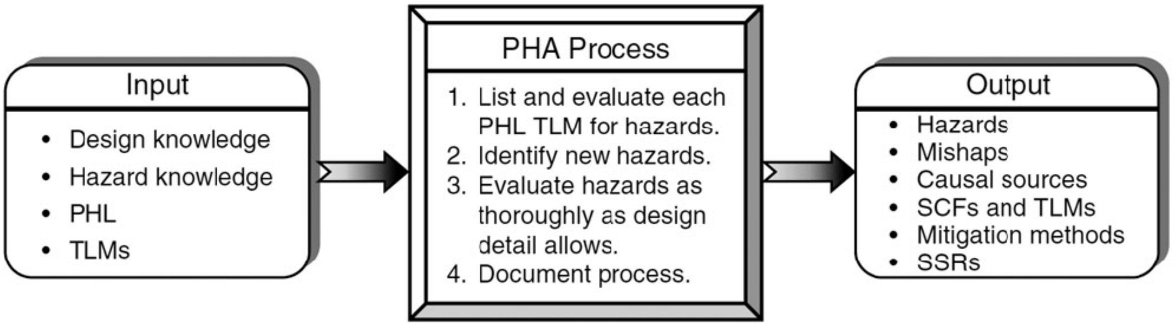


Figure 3. The PHA process according to Ericson.

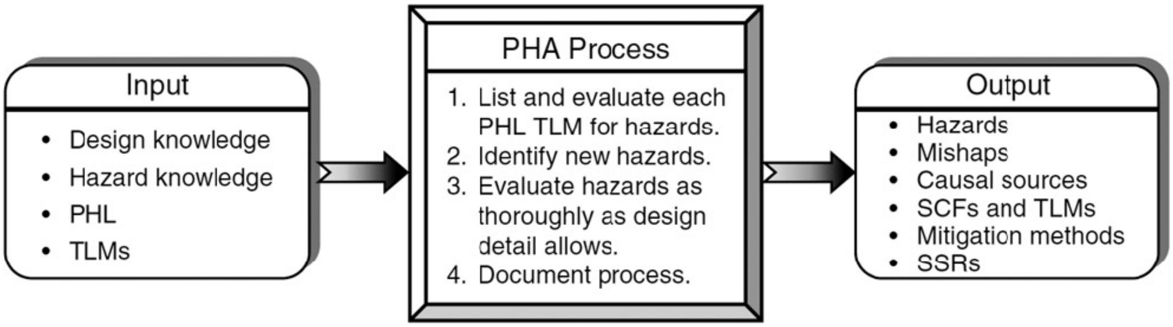


Figure 4. PHA Inputs, Process, and Outputs [2]

TLM- Top-level mishap
SCF- Safety critical functions
SSR- Systems safety reviews

PHAs again utilize worksheets and brainstorming sessions to accomplish their goals. Checklists similar to the one shown above are utilized to help guide the sessions. Though this approach has merit, it does not examine the entire system in a methodical way and, therefore, may miss hazards. “The underlying model of accident causation in the PHA literature is reductionist and assumes that if all… failure modes of a system have been identified, then so have all potential sources of hazardous behavior.” [5] In addition to hazards caused by failures and innately dangerous components, there are hazards that result from unforeseen component interactions, incorrect software design due to poor requirements, and human interaction errors that are a result of poor interfacing or system design. In modern airliners, only 20% of accidents are due to mechanical failures. [6] In complex systems, it is nearly impossible to foresee all the potential hazards without considering hazards that arise without a component failing. PHL and PHA do not systematically evaluate the system in a way that identifies hazards that arise due to non-failure events.

“The PHA (Figure 4) is perhaps the most critical analysis that will be performed because it is usually the first in-depth attempt to isolate the hazards of a new or, in some cases, modified system.” [7] Jeffrey Vincoli captures an important point in this statement. The early hazards analysis in a program set the stage for the entire systems safety process. Many of the later analysis draw on the hazards that are identified in the PHL and PHA. The system safety engineers set the stage for how they will interact with the rest of the team throughout the design of the project. Lastly, most of the major design tradeoffs are weighed and decided in the early stages of the project lifecycle. Based on a paper by Strafaci, Fleming developed the graph shown in Figure 5 to demonstrate how the cost of changing a system rises dramatically throughout the lifecycle of a program while the impact of the change decreases.

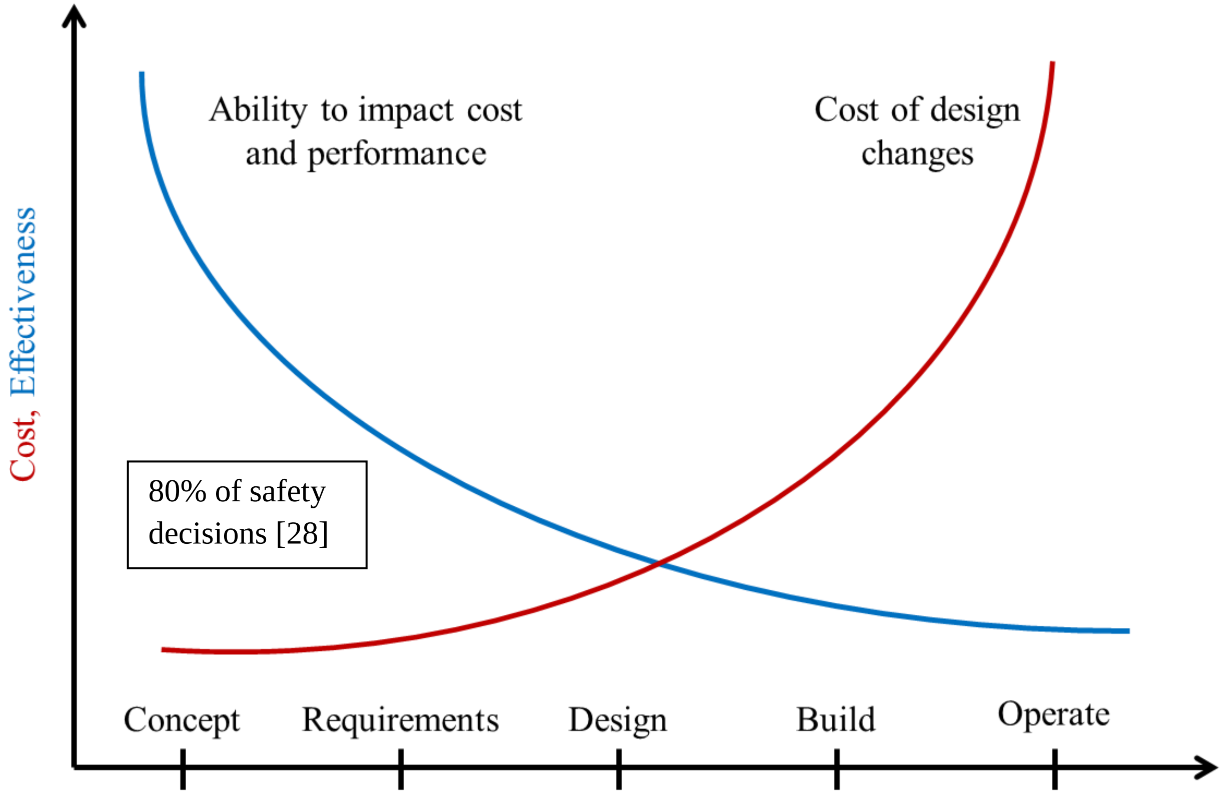


Figure 5. Decision Effectiveness during Life Cycle [5]

In their 1984 report, Frola and Miller reported that for defense projects, 70-80% of the decisions affecting safety are made in the early concept development stages of a project. With this in mind, it is incredibly important to invest in a robust systems safety analysis in the earliest stages of a project that can create a firm foundation for the rest of the effort and inform early safety decisions that will be costly or impossible to change later in the program. Furthermore, it is important to have robust safety information during tradeoff analysis so that safety can be adequately considered when making major design decisions.

In 2011, Leveson published *Engineering a Safer World*, a book outlining a new safety engineering approach that allows engineers to more fully analyze complex modern systems. In this work, Leveson proposes that safety should be treated as a control problem in which mishaps result from the interaction of dynamic processes in unplanned ways. Systems-Theoretic Accident Model and Processes (STAMP) is the accident model describing this concept. In STAMP, complex systems are modeled using control structures that show how constraints on the behavior of the system components are enforced. Safety is treated as an emergent property that is manifested when there is adequate control and feedback to enforce the desired behavior on the system. [8] This model captures hazards that result from component failures as well as hazards that arise from unsafe interactions between components in the system.

Systems-Theoretic Process Analysis (STPA) is an analysis technique based on STAMP. STPA guides an analyst through a structured method of examining a system and finding hazards and causal scenarios. The process of STPA is top-down and can therefore be applied from the earliest stages of concept definition and throughout the lifecycle of a system. STPA helps safety engineers identify potential causes of hazards and design a control structure that eliminates or mitigates them. STPA helps engineers find more hazards than a Preliminary Hazard List (PHL)and Preliminary Hazard Analysis (PHA) and guides engineers in finding ways to mitigate the hazards that are discovered.

In order to be used and accepted, STPA must not only provide good results but must also meet the requirements set in industry standards such as MIL-STD-882E. This effort is focused on hazard analysis during the concept phase of a program so tasks 201 and 202 will be the focus of the effort.

> ***TASK 201 – PRELIMINARY HAZARD LIST*** [4]
>
> *201.1 Purpose. Task 201 is to compile a list of potential hazards early in development.*
>
> *201.2 Task description. The contractor shall:*
>
> *201.2.1 Examine the system shortly after the materiel solution analysis begins and compile a Preliminary Hazard List (PHL) identifying potential hazards inherent in the concept.*
>
> *a. A brief description of the hazard.*
>
> *b. The causal factor(s) for each identified hazard.*

STPA starts with a list of mishaps and then defines high level hazards that the system faces. The high level hazards are decomposed into a detailed list of hazards in the system. From these hazards, STPA derives causal scenarios that describe the causal factors for each hazard. STPA meets the task description, providing a detailed process to follow to find hazards. STPA provides much more guidance and analytical power than PHL and enables engineers to find and list hazards in a methodical and intuitive way. STPA also satisfies task 202 in a manner that allows an easy transition from task 201.

> ***TASK 202 - PRELIMINARY HAZARD ANALYSIS*** [4]
>
> *202.1 Purpose. Task 202 is to perform and document a Preliminary Hazard Analysis (PHA) to identify hazards, assess the initial risks, and identify potential mitigation measures.*
>
> *202.2 Task description. The contractor shall perform and document a PHA to determine initial risk assessments of identified hazards. Hazards associated with the proposed design or function shall be evaluated for severity and probability based on the best available data, including mishap data (as accessible) from similar systems, legacy systems, and other lessons learned. Provisions, alternatives, and mitigation measures to eliminate hazards or reduce associated risk shall be included.*

As stated before, STPA is a structured method used to identify hazards and their associated risk to the system. Once an initial analysis has been completed, the STPA analysis can be refined as more design details are available. STPA can be used to determine severity of risk but does not assign probabilities of occurrence to hazards because of the issues listed earlier. If implemented by a qualified and experienced team like that required for a normal PHA, STPA will include input from similar and legacy systems. STPA leads very logically to defining mitigation measures to eliminate or reduce risks in a system. The rest of this work is devoted to demonstrating how STPA can be applied in the concept phase to perform a hazard analysis. STPA is used to evaluate the system being designed as well as the design process itself.

3. Theory: Systems-Theoretic Accident Modeling and Processes[1]

After becoming increasingly frustrated with the lack of progress in safety engineering and the mishaps that were being missed by traditional analysis techniques, about 12 years ago Leveson began developing a new, more comprehensive model of mishap causation based on system

[1] This section is adapted from an unpublished report written by Leveson and Horney for a contracted project. The work on systems theory and STAMP was done by Leveson.

theory. The model is called STAMP (System-Theoretic Accident Model and Processes).[2] Information about systems theory is provided first and then STAMP is described.

3.1 An Introduction to System Theory[3]

System theory dates from the 1940's and 1950's and was a response to the limitations of classic analysis techniques in coping with the increasingly complex systems that started to be built after World War II [9] [10]. Norbert Wiener [11] applied the ideas to control and communications engineering and called it *Cybernetics* while Ludwig von Bertalanffy [12] developed similar ideas in biology. Von Bertalanffy suggested that the emerging ideas in various fields could be combined into a general theory of systems, and that name (rather than Cybernetics) is more commonly used.

System theory was a reaction to the inability of traditional scientific and engineering approaches to handle the complexity of the new systems being built at the time, particularly defense systems such as the Intercontinental Ballistic Missile Systems (ICBM) and Early Warning Systems (EWS). System theory served as the theoretical foundation for System Engineering and System Safety, which were first developed for these complex defense systems [13] [14].[4]

In the traditional approach to dealing with complexity, sometimes referred to as *divide and conquer*, systems are divided up into distinct parts so that the parts can be examined separately and later the results of analyzing each separate component are combined to represent an analysis of the whole: Physical aspects of systems are decomposed into separate physical components while behavior is decomposed into discrete events over time. This can be represented as:

Physical aspects → separate physical (or sometimes functional) components
Behavior → discrete events over time

[2] For a detailed explanation of STAMP, see Nancy Leveson, *Engineering a Safer World*, MIT Press, 2012. A free .pdf version can be downloaded from https://mitpress.mit.edu/books/engineering-safer-world/

[3] Most of this section is adapted from [9], [14], and other papers and presentations by Leveson.

[4] For a short history of system safety, see Nancy Leveson, *Safeware: System Safety and Computers*, Addison-Wesley, 1995, pages 145-150.

This decomposition, formally called *analytic reduction*, assumes that such separation is feasible, that is:

- Each component or subsystem operates independently
- Analysis results are not distorted when the components are considered separately
    - Components or events are not subject to feedback loops and other nonlinear interactions
    - The behavior of the components is the same when examined singly as when they are playing their part in the whole.
- The principles governing the assembling of the components into the whole are straightforward, that is, the interactions among the subsystems are simple enough that they can be considered separate from the behavior of the subsystems themselves.

These are reasonable assumptions for many of the physical parts of the universe and for most pure electromechanical systems. Some system theorists have described these systems as displaying *organized simplicity* (Figure 6). Such systems can be separated into non-interacting subsystems for analysis purposes: the precise nature of the component interactions is known and interactions can be examined pairwise.

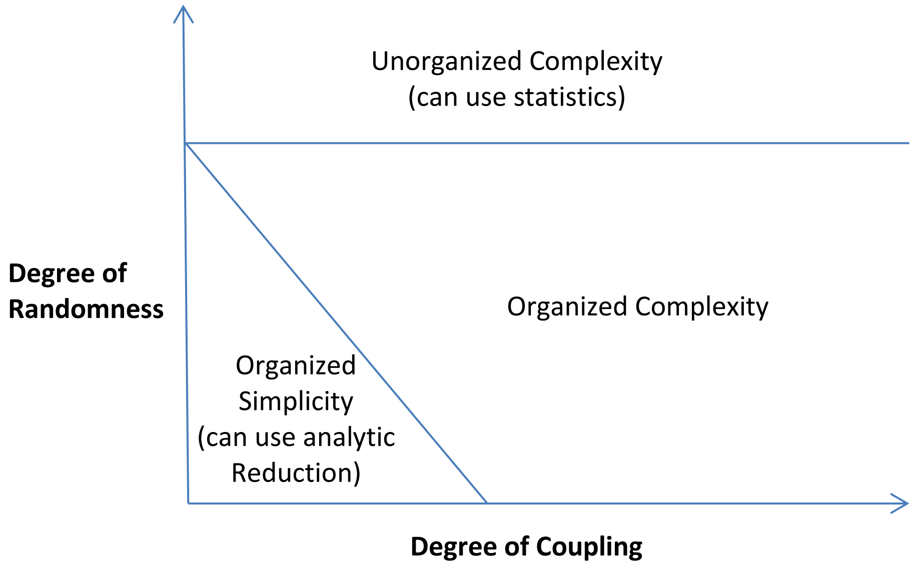


Figure 6.Three categories of systems (adapted from Gerald Weinberg, *An Introduction to General Systems Theory*, John Wiley, 1975)

Other types of systems display what system theorists have labeled *unorganized complexity*. They lack the underlying structure that allows reductionism to be effective. These types of systems can, however, often be treated as aggregates: They are complex, but regular and random enough in their behavior that they can be studied statistically. This study is simplified by treating them as a structureless mass with interchangeable parts and then describing them in terms of averages. The basis of this approach is the law of large numbers: The larger the population, the more likely that observed values are close to the predicted average values.

A third type of system exhibits what system theorists call *organized complexity*. These systems are too complex and the system components are too coupled for complete analysis and too organized for statistics; the averages are deranged by the underlying structure [15]. Many of the complex engineered systems of the post-World War II era, as well as biological systems and social systems, fit into this category. Organized complexity also represents particularly well the problems that are faced by those attempting to build complex software, and it explains the difficulty computer scientists have had in attempting to apply formal analysis and statistics to software.

System theory was developed for this third type of system. The system approach focuses on systems taken as a whole, not on the parts considered separately. It assumes that some properties of systems can only be treated adequately in their entirety, taking into account all facets and relating the social to the technical aspects [16]. These system properties derive from the relationships among the parts of systems (i.e., how the parts interact and fit together) [12].Thus the system approach concentrates on the analysis and design of the whole as distinct from the components or parts. This holistic approach provides a means for studying systems exhibiting organized complexity.

Emergence is a basic concept in system theory. Some properties in complex systems are emergent, that is, they are not just a “sum” of individual component behavior but arise in the interactions among the components. If the interactions are simple enough, that is, the behavior of one component has no or limited impact on the behavior of others (i.e., they are sufficiently decoupled), then the components can be analyzed separately and the analysis results combined to represent a sufficient approximation of the behavior of the whole. But as the interactions become more complex and coupled, emergent properties arise.

Figure 7 depicts the principle of emergence. A system or process is made up of components (i.e., the shaded boxes). The components interact in both direct and indirect ways. Emergent system or process properties arise from these interactions. The concept of emergence gives rise to the often quoted basic system theory principle that in complex systems "the whole is greater than the sum of the parts."

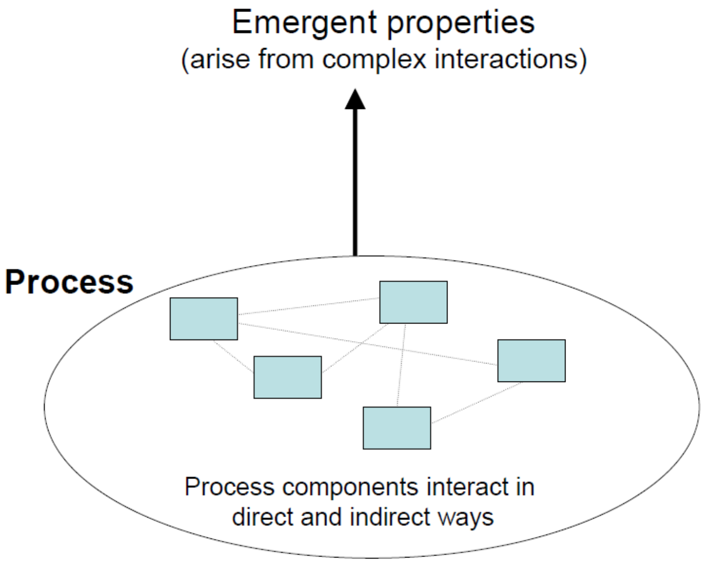

Emergent properties
(arise from complex interactions)
Process
Process components interact in
direct and indirect ways

Figure 7. Emergent properties arise from interactions among system components

Safety is an example of an emergent property, as is security and many other important system properties. Looking only at a "valve" and asking whether a system will be safe that uses that valve is an unanswerable question: System safety will depend on how the valve is used in a system and how it interacts with other components and in the operation of the whole. Note that the reliability of the valve as a component can be determined; but the "safety" of the valve, without consideration of its role in the whole, is limited in the hazards that can be considered by looking only at the valve, such as whether it has sharp edges by which a person could be injured.

Emergent properties associated with a set of components are, in system theory, related to constraints upon the degree of freedom of those components' behavior. In other words, to control emergent properties, the interactions among the components must be controlled in some way. Figure 8 depicts this principle. In an air traffic control system, allowing each aircraft to optimize its path will not optimize system properties such as collision avoidance and throughput. By enforcing constraints on the behavior of individual aircraft in the controlled airspace, collisions can be avoided and overall system throughput optimized.

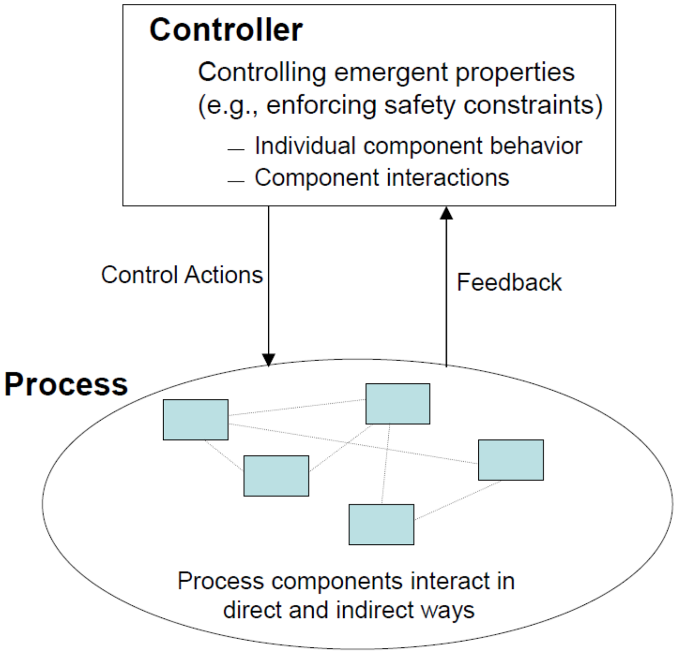

Controller
Controlling emergent properties
(e.g., enforcing safety constraints)
— Individual component behavior
— Component interactions
Control Actions
Feedback
Process
Process components interact in
direct and indirect ways

Figure 8. Safety reformulated as a control problem.

3.2 STAMP (System-Theoretic Accident Model and Processes)

Using system theory, System Safety is reformulated as a system *control* problem rather than simply a component *failure* problem:

Component Failure Problem

⇩

System Control Problem

In this formulation, mishaps or losses occur when component failures, external disturbances, and/or potentially unsafe interactions among system components are not handled or controlled adequately to enforce the required safety constraints (such as multiple aircraft maintaining minimum separation). Another way of saying this is that mishaps occur when the system safety constraints required to avoid hazards are not satisfied. Figure 8 illustrates that in order to prevent hazards and mishaps, safety constraints must be enforced on individual component behavior and on component interactions. This concept of control incorporates the basic control theory concept of a feedback control loop as shown above. Controls, however, can also be enforced through design features or other types of controls.

Controls can take a wide variety of forms and may be managerial, organizational, physical, operational, or manufacturing. Note that component failures are included here as a cause of mishaps, but additional causes not involving component failure are also considered. Also, standard forms of design features used to prevent or mitigate (control) component failures, such as redundancy may be used to prevent component failure-based hazards. The old model of accident causality is simply being extended by using system theory to include also the causes of accidents that arise in the interactions among components.

To summarize, in a system-theoretic view of safety, the emergent safety properties are controlled or enforced by a set of safety constraints related to the behavior of the system components. Safety constraints specify those relationships among system variables or components that constitute the non-hazardous or safe system states. If the hazard is two aircraft on a collision course, then the safety constraint is that aircraft must never be on a collision course or the paths must be corrected before a collision occurs. The safety constraints are just the inverse of the hazards, that is, the safety constraints are requirements that the hazardous conditions must be prevented from occurring or mitigated if they do occur. Mishaps (or losses) result from interactions among system components that violate those constraints—in other words, from a lack of enforcement of constraints on component and system behavior.

The controls that enforce the system safety constraints are embodied in the *hierarchical safety control structure*. Hierarchies are another basic concept in systems theory. At any given level of a hierarchical model of complex systems, it is often assumed that it is possible to describe and understand mathematically the behavior of individual components, i.e., that the component behavior is independent of other components at the same or other levels. But emergent system properties (such as safety) do not satisfy this assumption and require a description of the acceptable interactions among components. These interactions are controlled through the imposition of constraints upon the behavior of the components.

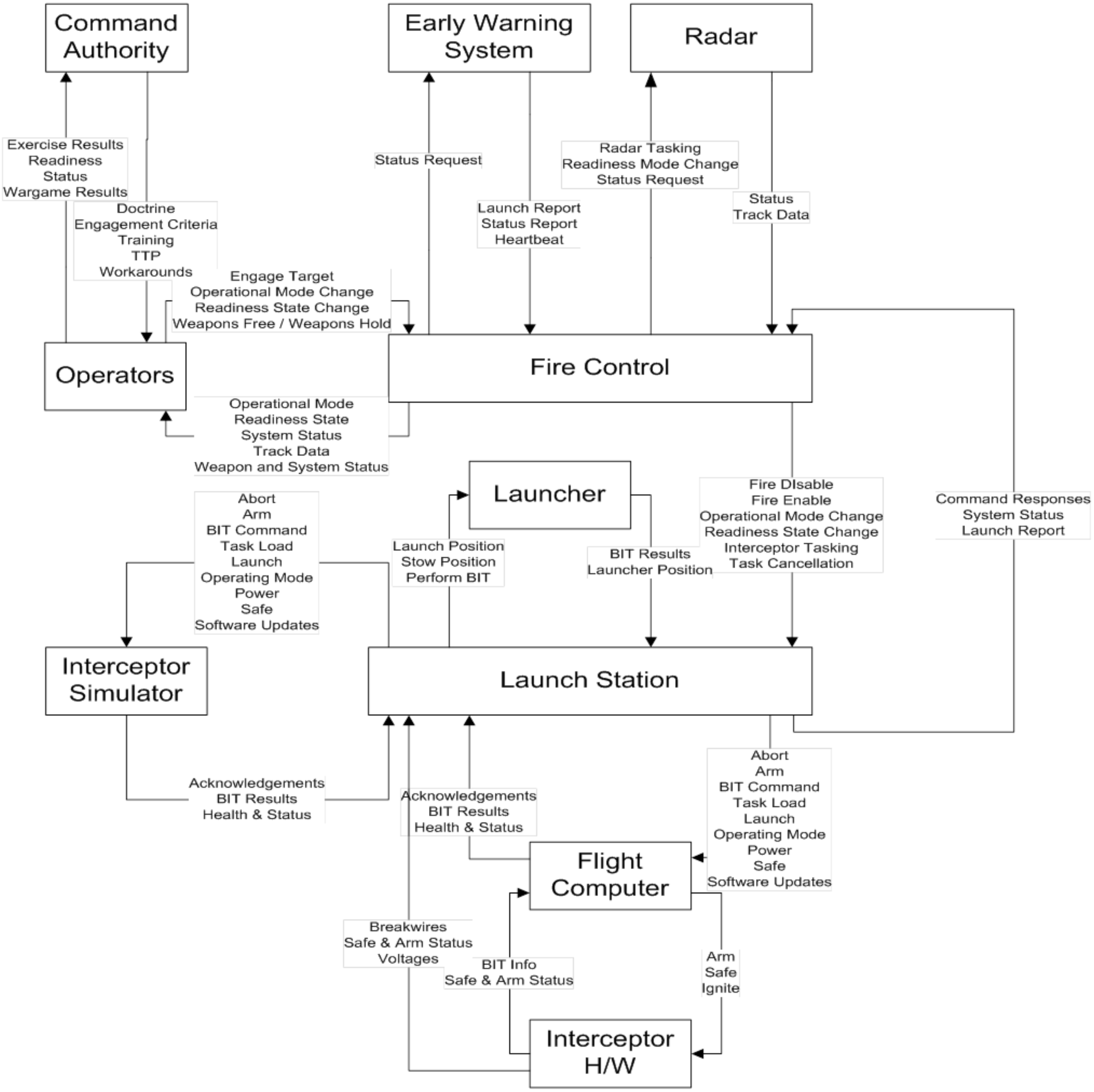


Figure 9. Safety Control Structure for a fictional Ballistic Missile Defense System. [17]

Figure 9 shows an example of a high-level hierarchical safety control structure for a fictional ballistic missile defense system [17]. Notice that it includes operators and the command authority and not just the hardware and software components in the system. In this model, the command authority controls the behaviour of the operators by providing doctrine, engagement criteria, training, etc. The operators control the behaviour of the Fire Control computer, which gets inputs from radars and the early warning system. The Fire Control computer provides instructions to and controls the Launch Station, which controls both the Launcher and the Flight Computer, and so on. Note that the safety control structure is not an architectural design model but a classic functional control structure.

In contrast, consider Figure 10 taken from SAE ARP 4761, which shows a classic functional decomposition (analytic reduction) of an aircraft.

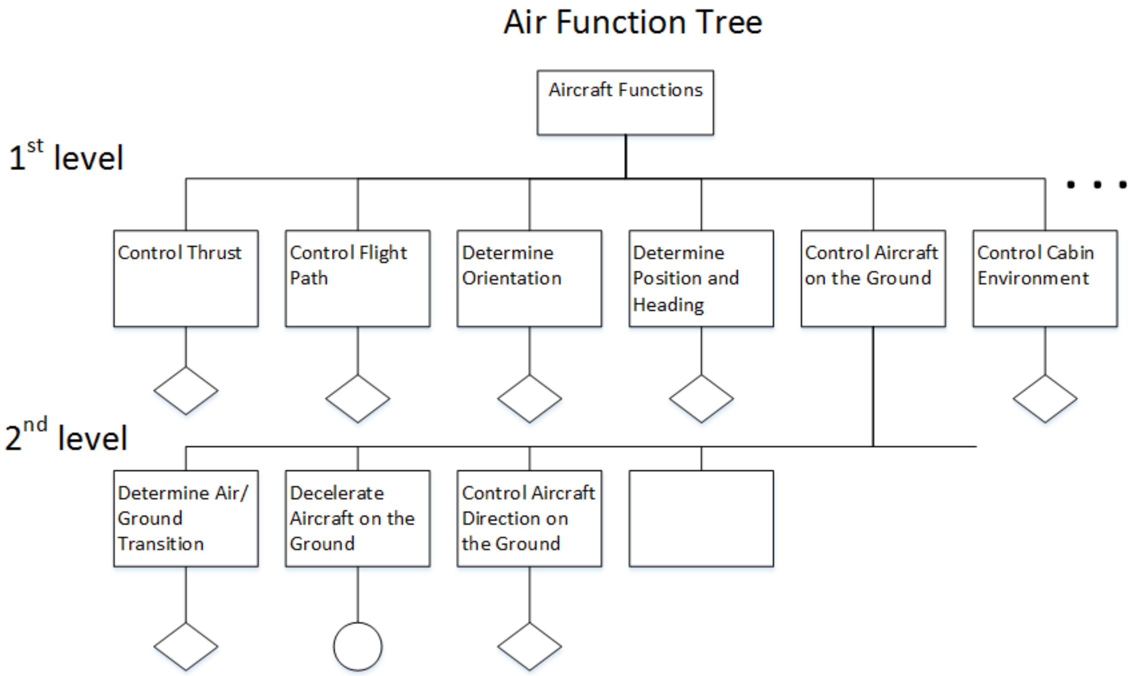

Air Function Tree
Aircraft Functions
1st level
Control Thrust
Control Flight Path
Determine Orientation
Determine Position and Heading
Control Aircraft on the Ground
Control Cabin Environment
• • •
2nd level
Determine Air/ Ground Transition
Decelerate Aircraft on the Ground
Control Aircraft Direction on the Ground

Figure 10.Two levels of a classic functional decomposition for an aircraft [18]

The analysis process in SAE ARP 4761 then involves calculating the probability of failure of each of the leaf nodes in the tree using Fault Tree Analysis (FTA) or other similar technique. They are then combined to get the probability of failure of the functions at the level above and so on until a probability of failure of the aircraft is derived.

In contrast, STPA starts with a control structure model. The mishaps and high-level hazards are first identified and then the control structure to prevent those hazards is modeled. For example:

Mishap-1. Loss of life or serious injury to aircraft passengers or people in the area of the aircraft

Mishap-2. Unacceptable damage to the aircraft or objects outside the aircraft

System-level hazards related to these losses include:

H1: Insufficient thrust to maintain controlled flight

H2: Loss of airframe integrity

H3: Controlled flight into terrain

H4: An aircraft on the ground comes too close to moving or stationary objects or inadvertently leaves the taxiway

The high-level control structure used by STPA for the aircraft is shown in Figure 11. The responsibilities assigned to each component (controller) in the safety control structure are modeled as well as the control actions available to the controller, the feedback the controller receives, and models of the controlled process used by the controller to select appropriate control actions. Note the difference between Figure 10 (representing classic analytic reduction) and Figure 11 and Figure 12 (representing system theory concepts). Figure 10 simply decomposes the required functions into separate boxes to be analyzed individually. Figure 11 and Figure 12 assign the functions to feedback control loops.

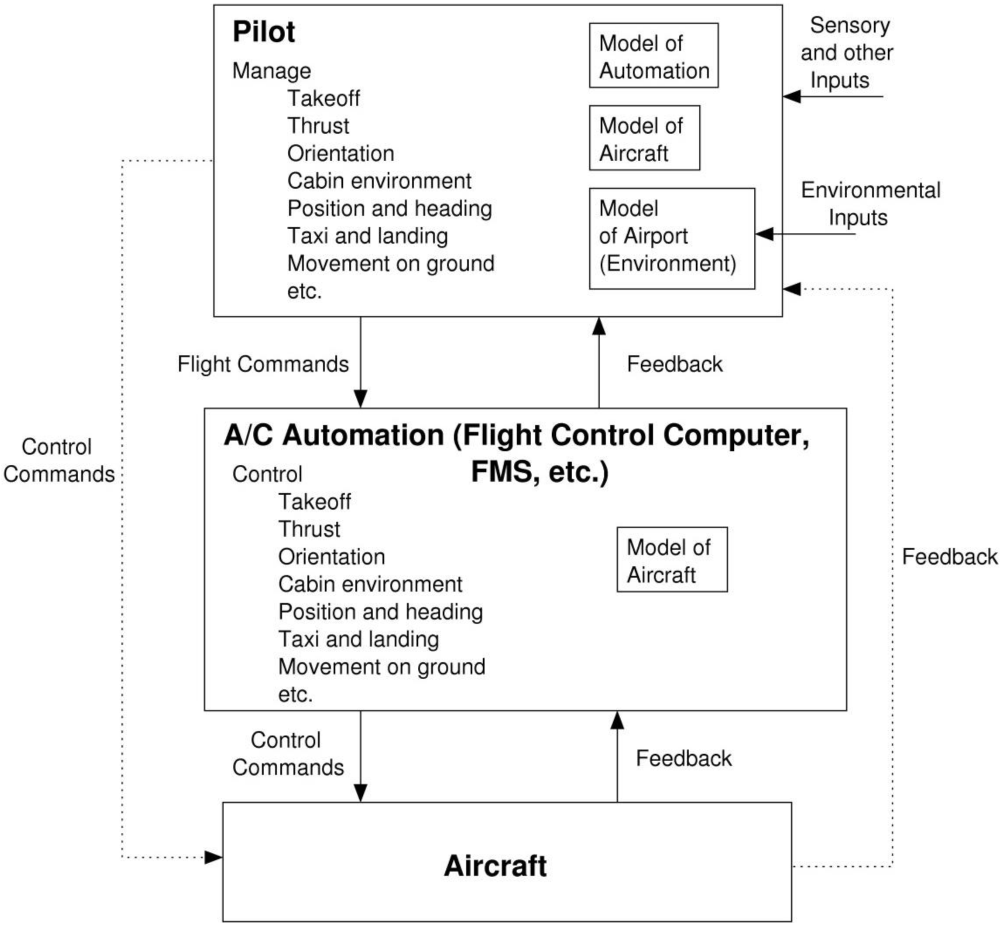


Figure 11.The high level control structure for an aircraft

As with the decomposition shown in Figure 10, the high-level control structure will be refined to be more detailed (an example for ground movement is shown in Figure 12). At an early stage of development, design details may not have been determined, but the safety constraints and causal scenarios for violating them can be identified for that level of refinement. These constraints (requirements) can then be refined when further design detail is created. If there is a need to back up to a higher level of design (e.g., the designers change their minds), it is easy to return to the higher-level of requirements generated by STPA and proceed from there.

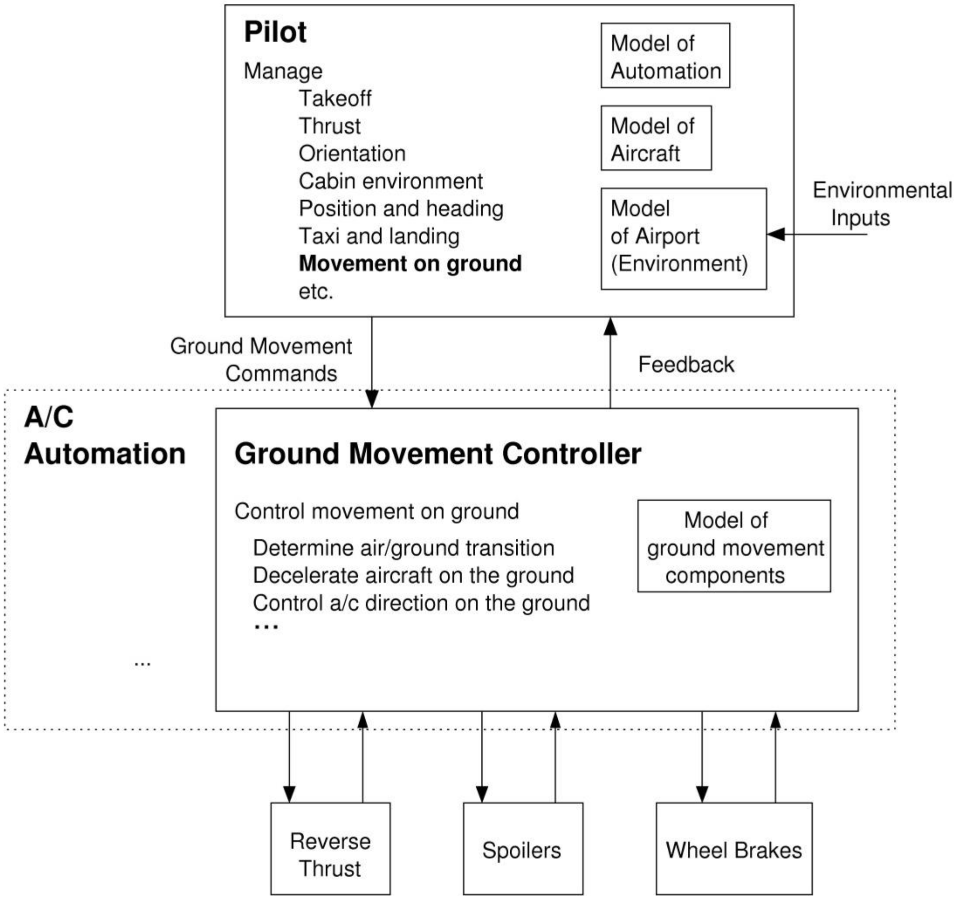


Figure 12.The portion of the safety control structure related to deceleration on the ground.

In system theory, every controller contains a model of the controlled process, called a *process model* in STAMP (Figure 13). For human controllers, this model is usually called a *mental model.* This process model or mental model includes assumptions about how the controlled process operates and the current state of the controlled process. It is used by the controller to determine what control actions are necessary to keep the system operating effectively and safely. A simple example is a thermostat that uses a model of the controlled space, including current temperature, desired temperature (set point), etc. to determine what actions to take to keep the temperature at the desired set point.

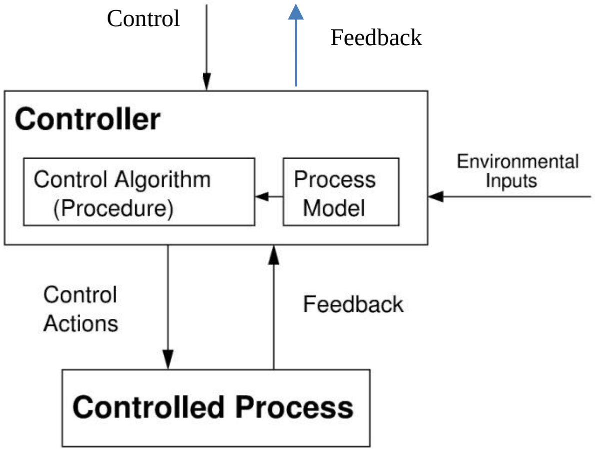


Figure 13. A typical control loop showing a process model

Mishaps in complex systems often result from inconsistencies between the process model used by the controller and the actual process state, which results in the controller providing unsafe control actions. For example, the autopilot software thinks the aircraft is climbing when it really is descending and applies the wrong control law; a military pilot thinks a friendly aircraft is hostile and shoots a missile at it; the software thinks the spacecraft has landed and turns off the descent engines prematurely; radar data is misinterpreted as an incoming threat and an interceptor is launched.

Part of the challenge in designing an effective safety control structure is providing the feedback and inputs necessary to keep the controller's model consistent with the actual state of the controlled process. As stated earlier, an important component in understanding mishaps involves determining how and why the controls are ineffective in enforcing the safety constraints on system behavior. Often a controller's unsafe behavior occurs because the process model used by the controller is incorrect. The required process models and feedback to maintain their consistency with the state of the controlled process are identified in the new hazard analysis techniques built on STAMP. The analysis methods identify the information that must be in the process model for safe control to occur and the requirements for updating it.

Inaccurate process models play an important role in a large number of mishaps involving software. Therefore, ensuring accurate information in the software process model is a critical part of the safe design of a system containing software.

The same is true for mishaps related to human errors. STAMP provides a potentially much more effective way of identifying safety-critical operator errors and their causes (so they can be eliminated or mitigated) than does treating human error like random machine failure, which is common in the traditional reliability approach to safety. For example, the control model allows more sophisticated analysis of human factors in mishaps including things like situation awareness flaws (i.e., a flawed process model), mode confusion, and distraction.

There are, in general, four ways that a controller can behave unsafely:

1. A provided control action leads to a hazard.
2. The lack of a control action leads to a hazard.
3. A potentially safe control action is provided too early, too late, or in the wrong sequence
4. A continuous control action is provided for too long or too short a duration

In addition, systems can get into hazardous states when a required control action is provided but not executed for some reason (e.g., a physical failure, a delay, etc.). These different ways that control can result in hazards form the basis for the new hazard analysis method called STPA (System Theoretic Process Analysis), which is performed on the system's safety control structure.

STPA is described with an example in section four, but briefly, the process involves:

1. Identifying high level mishaps and hazards
2. Constructing the safety control structure (including the potential control actions, feedback, and process models),
3. Identifying potential unsafe control actions,
4. Constructing the scenarios that could lead to the unsafe control actions.

Standard control theory concepts are used to construct the scenarios. Figure 14 shows some of the potential causes of unsafe control that may be involved in the scenarios.

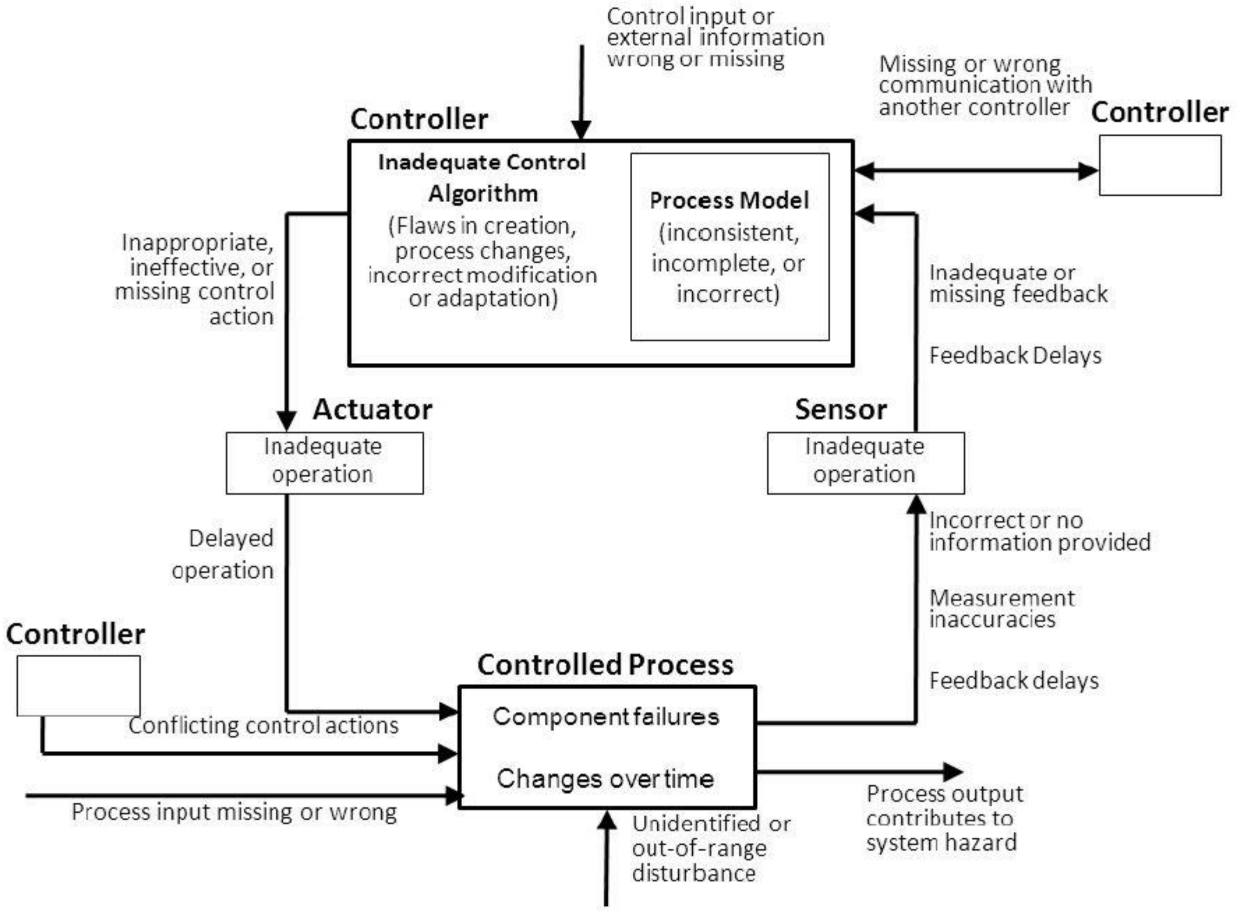

Control input or external information wrong or missing
Missing or wrong communication with another controller
Controller
Controller
Inadequate Control Algorithm
(Flaws in creation, process changes, incorrect modification or adaptation)
Process Model
(inconsistent, incomplete, or incorrect)
Inappropriate, ineffective, or missing control action
Inadequate or missing feedback
Feedback Delays
Actuator
Inadequate operation
Sensor
Inadequate operation
Delayed operation
Incorrect or no information provided
Measurement inaccuracies
Controller
Controlled Process
Feedback delays
Conflicting control actions
Component failures
Changes over time
Process input missing or wrong
Unidentified or out-of-range disturbance
Process output contributes to system hazard

Figure 14. Some control flaws that can lead to unsafe control. Note that component failures are included but control flaws other than component failures are also considered.

4. Safety Guided Design Using STPA

4.1 An Introduction to STPA and Safety Guided Design

The best time to start considering safety is at the very conception of a new project. STPA allows engineers to incorporate safety into the design process, which allows safety to be part of the tradeoff analysis and shape the system throughout the entire lifecycle. The integration of safety into the design process is called *safety-guided design.* [8] Figure 15 illustrates this process.

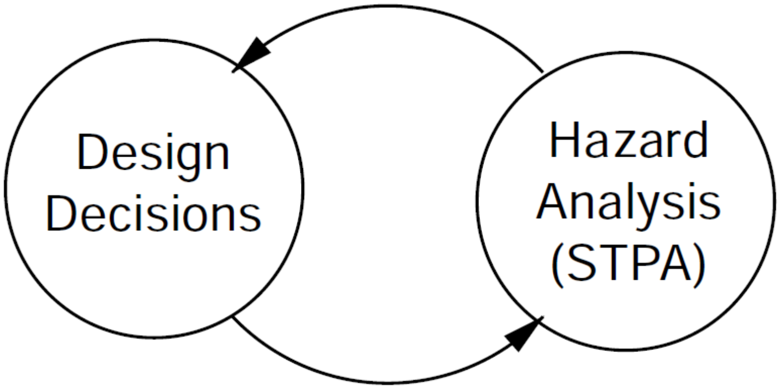


Figure 15. Safety-guided design entails tightly intertwining the design decisions and their analysis to support better decision making. [8]

Safety-guided design uses the normal STPA process to identify and mitigate hazards. In addition to performing an STPA analysis, engineers will perform additional design activity in between steps in the STPA process. Figure 16 shows the STPA process.

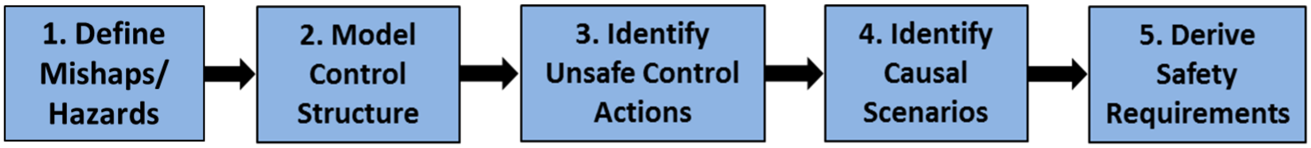

1. Define Mishaps/ Hazards
2. Model Control Structure
3. Identify Unsafe Control Actions
4. Identify Causal Scenarios
5. Derive Safety Requirements

Figure 16. STPA analysis process

1. Define Mishaps and Hazards.

Defining the mishaps for a program sets the scope for the rest of the safety activity. Typical mishaps include events that result in loss of life and or serious damage to the system in question or the surrounding environment. Depending on the nature of the system, other losses such as loss of mission or decrease in profits could also be considered. Defining the mishaps for a system sets the focus of the safety analysis. It is important to consult the stakeholders in the project when defining mishaps. A well thought through analysis will have the same core mishaps throughout the entire life of the program. Additional mishaps such as program cancellation can be considered during the appropriate design phases.

System level hazards are derived from the chosen mishaps and from safety criteria that is imposed by regulatory or industry associations and practices. [8] System level hazards lead directly to the defined mishaps. Very few systems have more than a dozen system level hazards. One way to test system hazards is to check and see if any of the hazards lead to each other. If a hazard leads to another hazard and not directly to a mishap it should not be included at the system level. Lower level hazards will be identified and considered in later steps of the analysis. After identifying the high level hazards, the design portion of safety-guided design begins. The first step to designing a safe system is to try to eliminate the system level hazards. If this can be done, there will be huge payoff in safety for the entirety of the project.

As an example, we will consider a hazard from the design of a computer network for a small company. The hazard is: **H1:** Inability to connect to the network. Network connectivity issues can be largely eliminated by building a wired network. This will make the network safer by decreasing the number of ways to reach a mishap such as **M1:** Loss of internet access. This example also illuminates that fact that safety becomes involved in the tradeoff analysis. In this case, the company could prefer a wireless network because it allows their employees to work on mobile computers anywhere in the office. This ability trades off against the reliability of a wired network. At times improving the safety has positive results in areas of performance and cost as well. Just like any other trade study, increasing the safety of a system has wide-ranging effects on other aspects of the system performance both positive and negative depending on the specific case.

2. Model Control Structure.

A system control structure is an incredibly valuable tool for any program. In addition to helping with the hazard analysis, the control structure provides a model for the design team of how the system will operate. Modeling the control makes many assumptions explicit and quickly points to deficiencies or inefficiencies in a design. Once the model is constructed, safety engineers must assign responsibilities to controllers for enforcing the system safety constraints that arise from the system level hazards. If multiple architectures are being considered, modeling the control structure for each possible design can help tremendously in the analysis of alternatives. A clearly modeled control structure helps unify the mental models of everyone on the design team. Having a shared mental model makes design and tradeoff analysis much more cohesive and smooth.

At the beginning of a project, it is important to limit the detail of the control structure to decisions that have already been made. It is easy to include details based on assumptions rather than those based on thought through design decisions. A principle that helps define the proper level of detail in the concept phase is to include only system components that are necessary for the basic functionality of the concept. Going back to the computer network example, the control structure would include the computers and the router but details about how they are connected would be omitted until design decisions are made.

3. Identify unsafe control actions (UCAs).

Once the system is modeled in a control structure, the control actions are defined and the detailed hazard analysis can begin. Each control action should be analyzed to find unsafe control actions. These unsafe control actions are the detailed hazards for the system. For a quick trade study, this could be the last step in the STPA process for a specific concept. Once the UCAs are identified, different concepts can be compared using these hazards.

4. Identify causal scenarios.

Starting with the unsafe control actions, causal scenarios that explain how the unsafe control actions can occur are identified. Once the potential causes of the hazards are known, engineers can alter the design to eliminate unsafe control actions and behaviors in the system. With every change to the system, the control model must be updated and the new control actions should be analyzed to find UCAs and causal scenarios. This iterative design process incorporates safety at the very beginning of the life of a system. After this process has been iterated and trade studies have been completed, a final concept can be decided upon and the remaining hazards can be studied to define safety requirements that will be implemented during the remaining design activities.

5. Derive Safety Requirements.

With each iteration, engineers will derive safety constraints and recommendations that shape the design. Once an architecture has been selected and the project moves towards requirements definition, the causal scenarios associated with the final design can be analyzed to drive safety requirements. These requirements will help illuminate or control the hazards that remain in the system.

Implementing this design process can take more time and resources than completing a PHL and PHA but the impact on the project is much greater. Integrating safety into the design of the system reduces the need for rework and makes the implementation of safety related design features easier and less expensive throughout the life of the system. Furthermore, a safer design will have fewer mishaps throughout its lifecycle, which reduces cost and improves performance. Using STPA for safety-guided design allows safety to be seamlessly integrated into the design and trade process from the very inception of a project.

Safety-Guided Design of Light Military Transport

4.2 Define mishaps and hazards

To begin the STPA hazard analysis, the mishaps to be considered are defined. For this analysis, three broad mishaps have been identified as a focus. These mishaps define the unacceptable losses for the hazard analysis.

Mishap 1: Serious injury or fatality to personnel
Mishap 2: Loss of or damage to the aircraft or equipment on the aircraft
Mishap 3: Inability to complete the mission

This portion of the analysis is focused around designing the aircraft to operate safely. Other portions will focus on different mishaps related with operations. Later, additional safety requirements can be generated for specific missions and capabilities, which may expand the mishaps considered. Any newly added safety requirements, must of course, be shown to be consistent with existing safety requirements. A top-down process assists in this task.

After the mishaps to be considered are identified (and validated by the stakeholders), the standard process is to identify the system hazards that can lead to the mishaps. Again, the following hazards are very high level and are applicable to the entire system. Lower-level (more specific) hazards and more detailed general and mission-specific hazards are identified in the STPA analysis process as will be shown. Each hazard is traced back to the mishap that it could cause. In this case, each hazard could potentially lead to any or all of the three mishaps that have been identified. The hazards can be rewritten as constraints that must be satisfied by the aircraft design and operations. By starting at a high-level and essentially building a tree of more detailed hazards as they are refined, the omissions, redundancies, etc. can be identified more easily. As will be seen below, STPA has a step-by-step process for identifying the more detailed hazards from these very high-level hazards. The detailed hazards are traced to the high-level hazards.

Hazard	**Constraint**
H1: Violation of minimum separation standards (M1, 2, 3)[5]	The aircraft must maintain minimum separation from potential sources of collision.
H2: Inability to control the aircraft (M1, 2, 3)	The aircraft must be controllable by the pilot or piloting function in an OPV (optionally piloted vehicle) at all times.
H3: Loss of airframe integrity (M1, 2, 3)	Airframe integrity must not be lost during flight.

The high-level hazards and mishaps identify and define the goal for the analysis, which further refines the hazards and identifies causal scenarios that can lead to them. The causal scenarios can be used to provide guidance for the designers to eliminate or mitigate hazards in the design process. As design decisions are made, the analysis can be iterated and refined.

The high level hazards are now evaluated before moving on. Is it possible to eliminate any of the hazards by changing the concept in a way that still allows the mission to be completed? The first hazard, violation of separation is present in any system that moves. This mission requires the transport of people and cargo and thus violation of separation will always be a hazard in the system. In order to minimize the effects of a violation of separation, energy should be managed. Colliding with an object at a high energy state results in greater damage. For aircraft, this principle is slightly reversed. It is safest to fly at higher altitudes and speeds far above stall. This gives controllers more time and energy to use if something goes wrong. The aircraft should be designed to operate at an altitude that gives the controller time to react to an emergency and in attitudes that are well within the flight envelope of the airframe. This will help minimize the damage caused by a loss of separation. Good feedback about the environment also helps reduce the hazards of loss of separation. This aircraft should be able to sense any objects around it that could damage it with enough warning to avoid the object. Even at the highest level,

[5] This annotation traces the hazards back to the mishaps to the related mishaps. “M1” refers to Mishap 1, “Loss of or damage to the aircraft or equipment on the aircraft” etc.

this design method focuses designers on the safety of the system and how design decisions will impact the safety of the system.

Is it possible to eliminate the possibility of losing control? Although control algorithms, autopilots, and control redundancy has improved greatly over the past decades, it is still possible for any aircraft to lose control. Again, mitigating the likelihood and effects of losing control becomes the goal. The same principles of maintaining altitude and flight well within the flight regime help mitigate the dangers of losing control assuming it can be regained. Designing a stable airframe also makes it more difficult to lose control. For some missions, stability is detrimental to handling so a tradeoff must be made. For a fighter mission, it would be safer for the aircraft to be more maneuverable and able to defeat its foe than for it to be stable in case of a loss of the control computer. This demonstrates that safety is related to the mission as a whole rather than just emergency management. For combat aircraft, safety is closely coupled with mission effectiveness.

Lastly, consider the loss of airframe integrity. Due to the energy of flight and the potential for high loads and combat damage, this hazard is also unavoidable under certain circumstances. To avoid loss of airframe integrity, the aircraft should avoid bad weather, enemy contact, and the components should be monitored throughout the lifecycle. It is rare to be able to eliminate a high level hazard for a system like an aircraft that operates at high energy states. Safety-guided design is able to point engineers towards designs that minimize the hazards from the highest level information. This gets engineers thinking about safety almost immediately in and throughout a project, which is incredibly valuable.

4.3 Model Control Structure

STPA views safety primarily as a control problem. Mishaps occur when the safety constraints are not enforced in the design and operation of the system. STPA is applied to a control structure model. Various levels of abstraction and granularity may be reflected in different or models of the overall safety control structure, resulting in different types of results from STPA. During the concept stage, it is possible to define the control structure in multiple ways. As the design becomes more concrete, the control structure will become more defined and, thus, a need to iterate on the STPA process. Section 5 demonstrates how STPA can be used in a tradeoff analysis that investigates multiple ways of achieving the design objectives.

As an example, the control structure shown in Figure 17 models the general U.S. Army, a potential user of the light transport, command structure and demonstrates how safety constraints are enforced in the Army as an organization. Typically each commander's staff will include trained safety officers that have intimate knowledge of the Army's safety programs and systems (shown in the box behind each commander). This control structure can be analyzed using STPA to ensure that safety related responsibilities are properly defined within the organization.

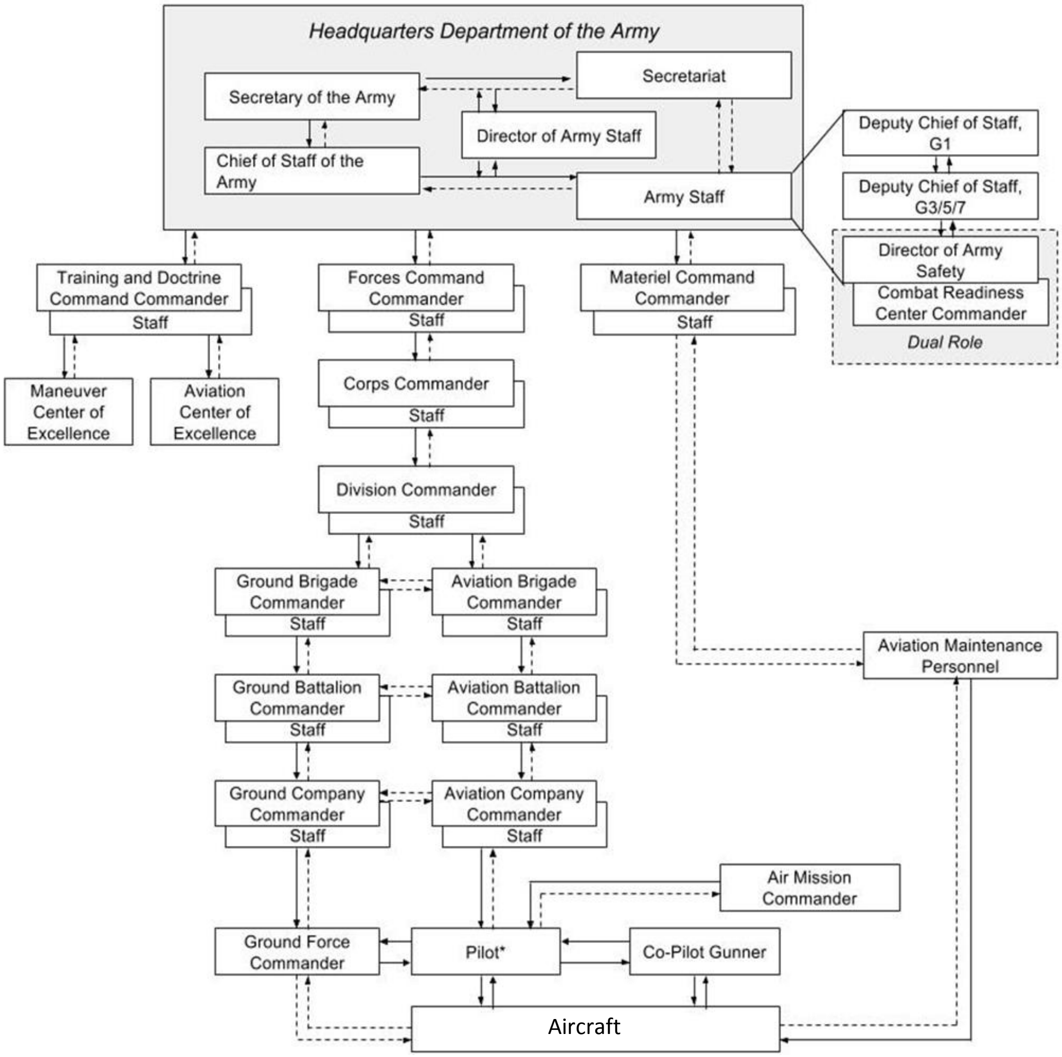

Headquarters Department of the Army
Secretary of the Army
Secretariat
Director of Army Staff
Chief of Staff of the Army
Army Staff
Deputy Chief of Staff, G1
Deputy Chief of Staff, G3/5/7
Director of Army Safety
Combat Readiness Center Commander
Dual Role
Training and Doctrine Command Commander
Staff
Forces Command Commander
Staff
Materiel Command Commander
Staff
Maneuver Center of Excellence
Aviation Center of Excellence
Corps Commander
Staff
Division Commander
Staff
Ground Brigade Commander
Staff
Aviation Brigade Commander
Staff
Ground Battalion Commander
Staff
Aviation Battalion Commander
Staff
Ground Company Commander
Staff
Aviation Company Commander
Staff
Aviation Maintenance Personnel
Air Mission Commander
Ground Force Commander
Pilot*
Co-Pilot Gunner
Aircraft

Figure 17. U.S. Army high-level safety control structure. [19]

A more detailed model of the Materiel Command could be used in analyzing potential "hazards" in the engineering development and acquisition processes for a program. Other models focusing on parts of the system directly connected to operational command of the aircraft may also be useful in a sociotechnical analysis. Figure 18 Operational safety control structure example zooms in on the organizations involved in the operational actions of conducting an engagement. It shows the relationships with respect to controlling safety between different entities in the tactical environment.

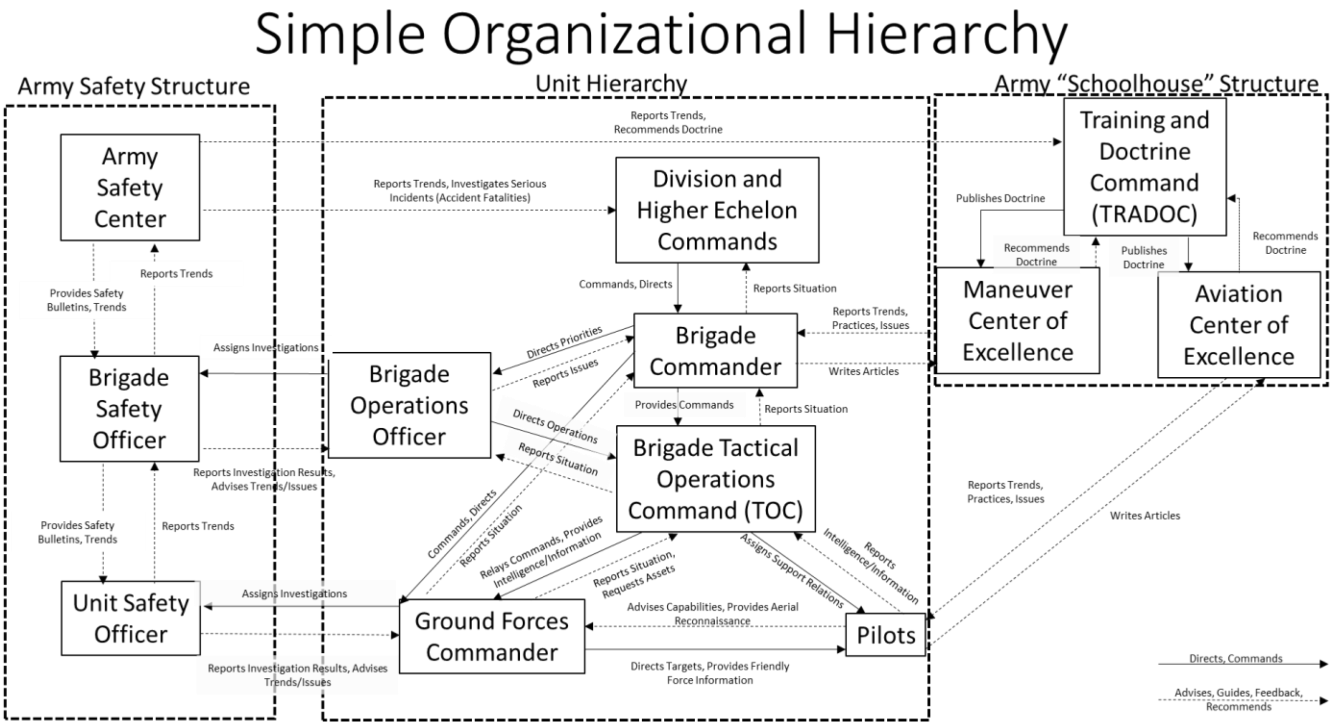

Simple Organizational Hierarchy
Army Safety Structure
Unit Hierarchy
Army "Schoolhouse" Structure
Army Safety Center
Reports Trends, Recommends Doctrine
Reports Trends, Investigates Serious Incidents (Accident Fatalities)
Division and Higher Echelon Commands
Training and Doctrine Command (TRADOC)
Publishes Doctrine
Recommends Doctrine
Publishes Doctrine
Recommends Doctrine
Maneuver Center of Excellence
Aviation Center of Excellence
Provides Safety Bulletins, Trends
Reports Trends
Commands, Directs
Reports Situation
Reports Trends, Practices, Issues
Writes Articles
Assigns Investigations
Brigade Safety Officer
Brigade Operations Officer
Directs Priorities
Reports Issues
Brigade Commander
Provides Commands
Reports Situation
Directs Operations
Reports Situation
Brigade Tactical Operations Command (TOC)
Reports Investigation Results, Advises Trends/Issues
Provides Safety Bulletins, Trends
Reports Trends
Commands, Directs
Reports Situation
Relays Commands, Provides Intelligence/Information
Reports Situation, Requests Assets
Assigns Support Relations
Reports Intelligence/Information
Reports Trends, Practices, Issues
Writes Articles
Assigns Investigations
Unit Safety Officer
Ground Forces Commander
Advises Capabilities, Provides Aerial Reconnaissance
Pilots
Reports Investigation Results, Advises Trends/Issues
Directs Targets, Provides Friendly Force Information
Directs, Commands
Advises, Guides, Feedback, Recommends

Figure 18 Operational safety control structure example [19]

Only the aircraft and pilots are included in this STPA demonstration. However, the hazard causal scenarios for the aircraft systems can be used to generate organizational safety requirements, particularly for hazards where the causal scenarios cannot be completely eliminated or adequately mitigated in the physical design of the aircraft and must be controlled during operations. For example, new training requirements for aircraft may be generated for hazards related to the new technology introduced or there may be new requirements/constraints on the conditions under which missions are planned and executed.

Examples of STPA applied to aviation system organizational structures and safety management systems can be found in Stringfellow [2010], who applied STPA to the FAA control over unmanned aircraft, and in Chung [2014], who used STPA for an analysis of the Air Force Test Center safety management system.

Figure 19 resumes the focus of the present analysis. It shows a high-level model of the flight vehicle control structure that serves as the focus for the analysis and provides the basis for identifying safety and security concerns that can be addressed during design. As design progresses, STPA should be iteratively applied to reveal more detailed safety constraints to guide the design decision making process.

The model in Figure 19 includes general responsibilities and control relationships that represent the light transport concept. Modeling is kept as generic as possible during the early design process in order to promote commonality and interoperability in any resulting model of the functional architecture. Additions may be made, such as additional communication between aircraft systems, but it is unlikely that anything will be eliminated. The goal is to provide a control structure that can serve as a starting point for conceptual design and that can be augmented and refined in future design activities.

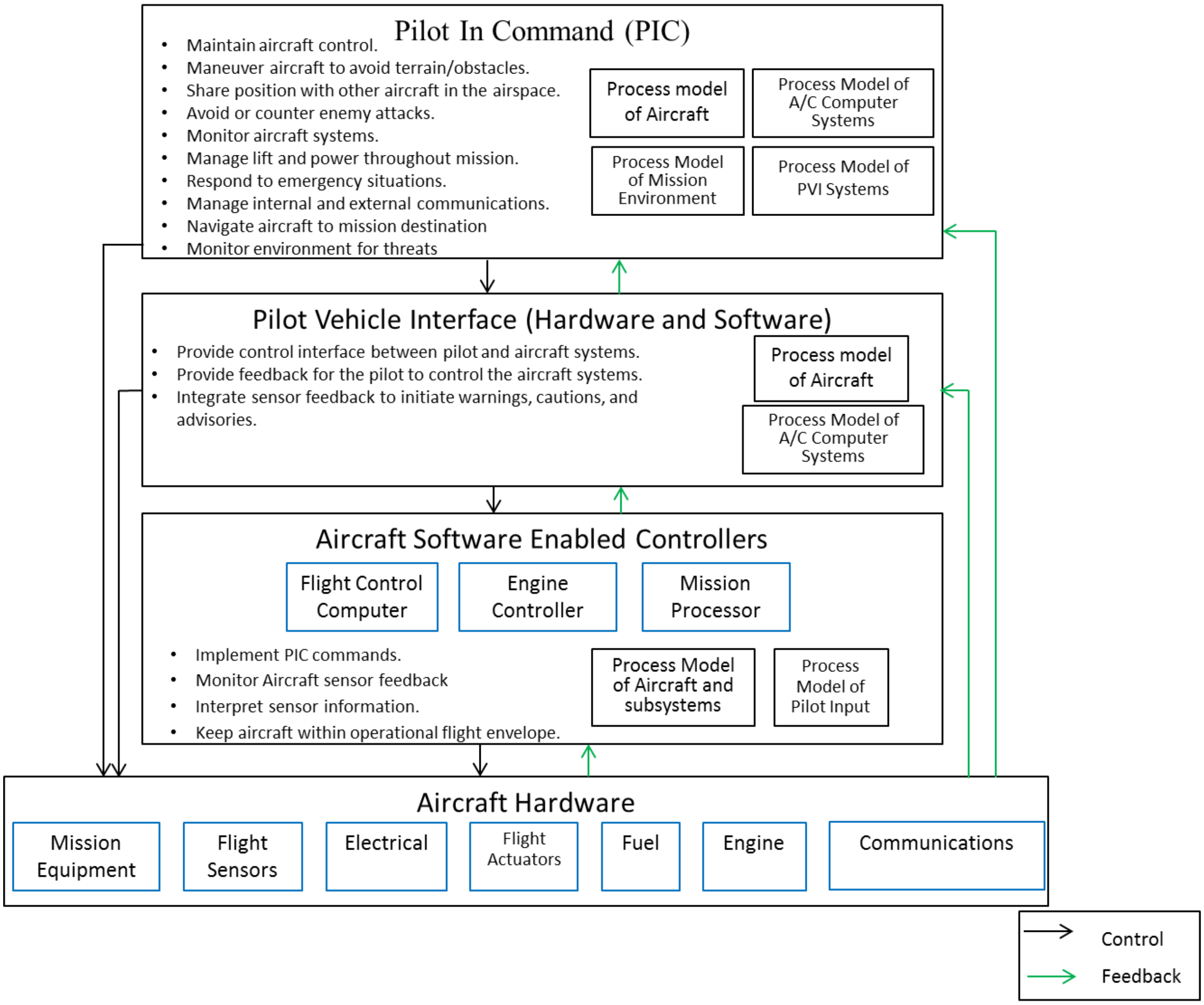

Pilot In Command (PIC)
Maintain aircraft control.
Maneuver aircraft to avoid terrain/obstacles.
Share position with other aircraft in the airspace.
Avoid or counter enemy attacks.
Monitor aircraft systems.
Manage lift and power throughout mission.
Respond to emergency situations.
Manage internal and external communications.
Navigate aircraft to mission destination
Monitor environment for threats
Process model of Aircraft
Process Model of A/C Computer Systems
Process Model of Mission Environment
Process Model of PVI Systems
Pilot Vehicle Interface (Hardware and Software)
Provide control interface between pilot and aircraft systems.
Provide feedback for the pilot to control the aircraft systems.
Integrate sensor feedback to initiate warnings, cautions, and advisories.
Process model of Aircraft
Process Model of A/C Computer Systems
Aircraft Software Enabled Controllers
Flight Control Computer
Engine Controller
Mission Processor
Implement PIC commands.
Monitor Aircraft sensor feedback
Interpret sensor information.
Keep aircraft within operational flight envelope.
Process Model of Aircraft and subsystems
Process Model of Pilot Input
Aircraft Hardware
Mission Equipment
Flight Sensors
Electrical
Flight Actuators
Fuel
Engine
Communications
Control
Feedback

Figure 19 High-level aircraft control structure

The model in Figure 19 requires some explanation. First, the model is a functional model, not a physical one. It shows generic control relationships and not necessarily physical relationships. For example, the Pilot-in-Command (PIC) controller shown does not imply that there is a human pilot in the airborne aircraft or that the PIC is even human. The PIC may be on the ground or in another aircraft (in a tethering situation). In the future, the PIC may be completely automated with no real-time human piloting control (as is true for some drones today). Human input into the piloting function in the completely automated design might only involve prior programming of the piloting function. In most near-term designs and even future designs, there will be partial automation, where the functions performed today by a human pilot are implemented in software with a human providing only oversight and monitoring. There is not meant to be any implication in the model that the functions are all implemented by a human pilot but only that the general functions must be implemented in some way. The safety constraints/requirements generated from the STPA analysis will apply to whatever system components (human, software, or hardware) implement the functions.

Second, the hierarchical level of control labeled "Aircraft hardware" may be composed of subsystems for each of the general hardware components. For example, the fuel system will probably have a (software) controller and be composed of several subsystems and controllers of these subsystems. These details will be considered in later refinements of the control structure. This top-level structure is meant to only show the general relationships.

More detailed versions of the control structure are used to specify the details of the model. The more detailed control actions and feedback relationships in Figure 19 are shown in Figure 20- Figure 22 and the control details provided. The complete list of labels for the control structures are in Appendix A. Control Diagrams with Labels

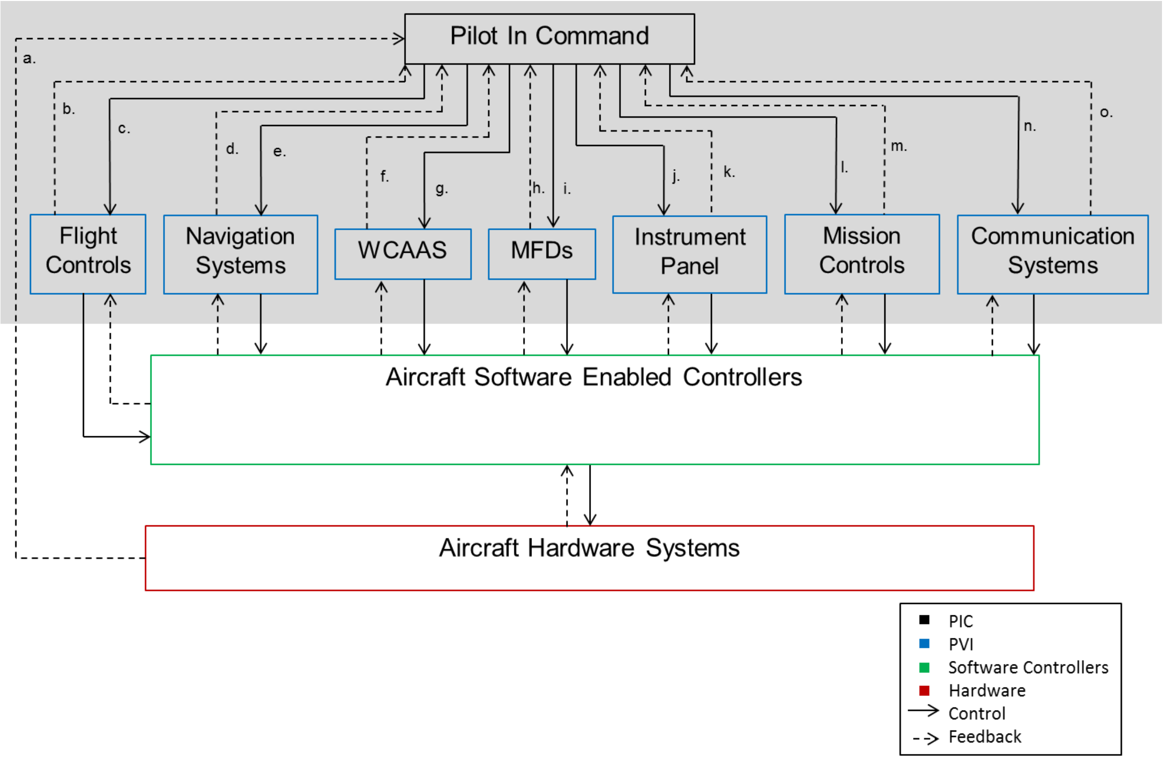


Figure 20. Details of the interactions of the PIC with the different parts of the Pilot Vehicle Interface (PVI) (hardware and software). To make the model more readable, only the details of the PIC and PVI are shown, with the other boxes providing overall context. The blue boxes represent physical interfaces as well as embedded software.

a. Feedback: Aircraft Hardware Systems → PIC

Noise from the airframe

Engine noise

Vibrations

Visible battle damage

Hardware status

Odors from aircraft systems

Environmental and system conditions (such as temperature and pressure of the cabin, temperature of the avionics bay, fire and smoke indicators)

b. Feedback: Flight Controls → PIC

Haptic flight control feedback

Flight control position

c. Control Actions: PIC → Flight Controls

Set aircraft attitude

Set aircraft power

Set aircraft altitude

Set formation shape

Choose emergency response for tethered vehicles

Etc. …

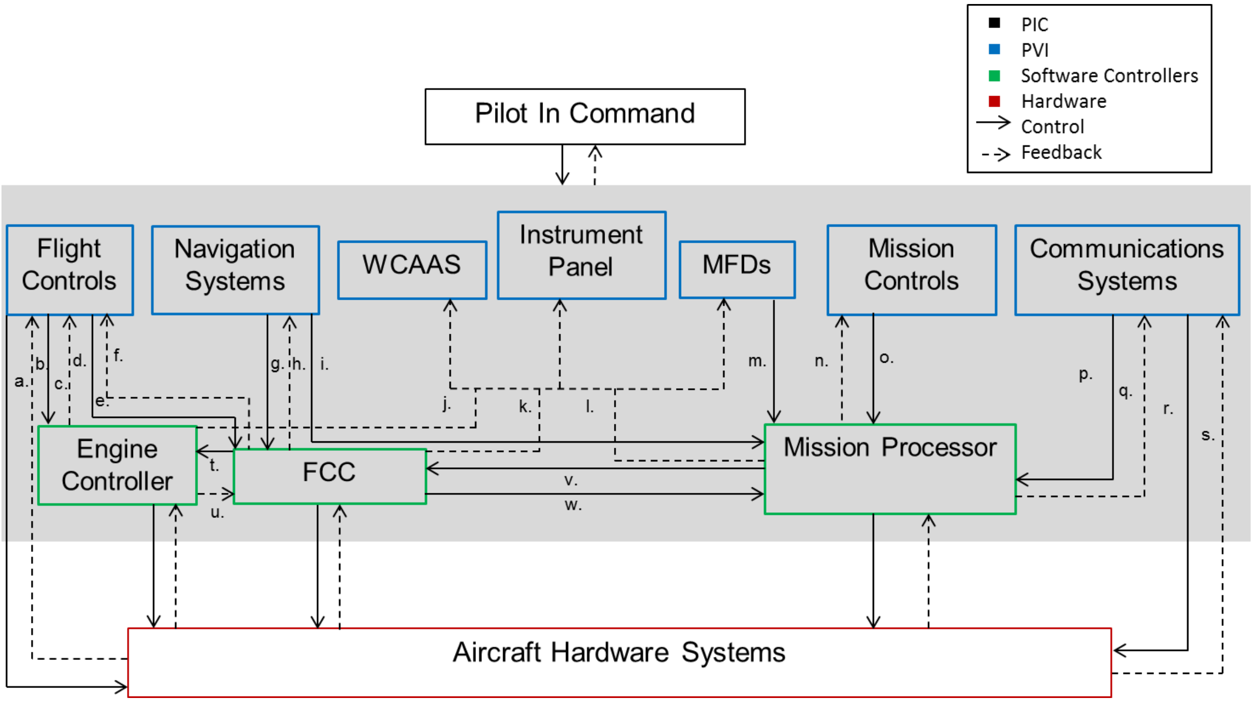


Figure 21. Detailed control structure for Pilot Vehicle Interface (hardware and software) to Aircraft Software-Enabled Controllers (shown in the grey area).

a. **Control Actions: Flight Controls → Aircraft Hardware Systems**

   Actuate directly connected flight systems

b. **Feedback: Aircraft Hardware Systems → Flight Controls**

   State of directly controlled hardware

c. **Control Actions: Flight Controls→ Engine Controller**

   Translate engine control inputs to be implemented by the engine controller

   Etc. …

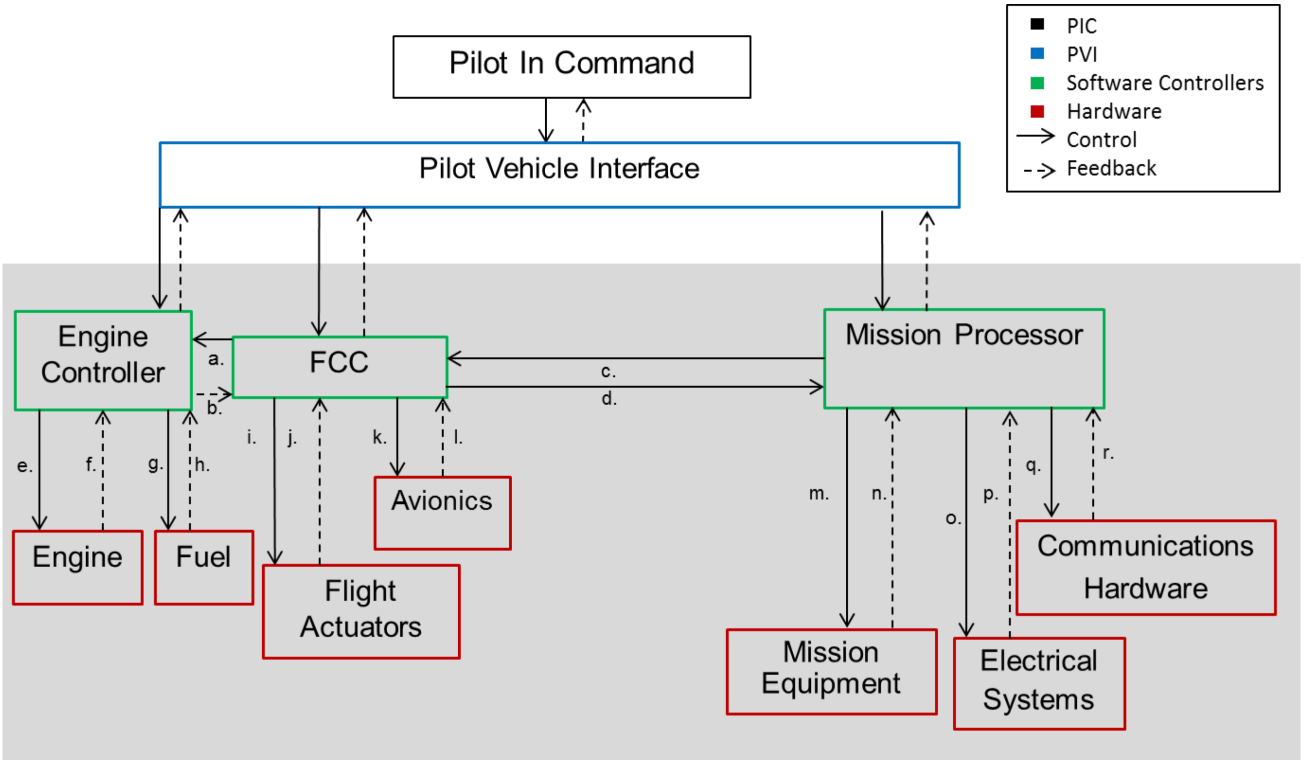


Figure 22. Detailed control structure for Aircraft Software-Enabled Controllers to Aircraft Hardware systems.

a. **Control Actions: FCC → Engine Controller**

Desired power

Atmospheric Information

b. **Feedback: Engine Controller → FCC, Mission Processor**

Engine RPM

Engine temperatures

Maximum performance capabilities

Power output

Power available

c. **Feedback: Mission Processor → FCC, Engine Controller**

Weight and balance information

Etc.…

These diagrams include a functional abstraction of the systems necessary to control an aircraft. The diagrams highlight the features in the pilot vehicle interface (PVI) that are necessary for a manned vehicle as this concept calls mainly for manned missions. The tethered aircraft would have the same equipment available but would not use it during tethered flight. The control diagram for the following tethered aircraft is the same except that the software enabled controllers are receiving commands through a data link from the lead aircraft rather than their native PVI. This fact will be highlighted during the formation of unsafe control actions. Some of the details listed above are assumed based on previous aircraft. For example, it is assumed that the aircraft will have a Warning Caution Advisory Annunciation System (WCAAS). This is assumed based on the presence of these systems in nearly all modern aircraft. The function is important to a safe aircraft and thus it is included although not defined in the concept specifically. During concept formation, there will be certain components that are contained in every concept proposed and such components can be included in the earliest hazard analyses in order to make a more complete model of the system.

4.4 Identify Unsafe Control Actions

After modeling the control structure of the parts of the system to be incorporated in the analysis, the next step of STPA is to define the conditions under which control actions in the aircraft can be unsafe. An unsafe control action consists of four parts:

(1) The *controller* issuing the control action;

(2) The *type* of control action (providing can lead to the hazard, not providing can lead to the hazard, incorrect order or timing leads to the hazard, or incorrect duration for a continuous (vs. discrete) control action);

(3) The *control action* itself; and

(4) The *context* under which the control action becomes hazardous.

For example, “The PIC [*controller*] does not [*type*] set aircraft pitch [*control action*] when the trimmed aircraft state is causing the aircraft to deviate from its flight plan [*context*].”

The examples used in this section consider control actions by the pilot in command (PIC) and by the Fight Control Computer (FCC). The first PIC action considered is “set aircraft pitch”.

Table 1. PIC to Flight Controls UCAs shows the control actions by the PIC in the left column. The other four columns describe the conditions under which these control actions can be hazardous (i.e., Not Providing Causes Hazard, Providing Causes Hazard, Incorrect Timing/Incorrect Order, Stopped Too Soon/ Applied Too Long). Note that the descriptions of unsafe control actions in the table are the usual, more detailed hazards identified in a traditional hazard analysis. STPA provides structured guidance to generate these hazards to ensure completeness. In fact, the unsafe control actions (detailed hazards) can be generated automatically using techniques developed by John Thomas at MIT [20]

At the end of each Unsafe Control Action (UCA), there is a reference to the related hazards. For example, UCA 1.1 in Table 1 can result in a violation of minimum separation, which is hazard H1.

Table 1. PIC to Flight Controls UCAs

Control Action	**Not Providing Causes Hazard**	**Providing Causes Hazard**	**Incorrect Timing/ Incorrect Order**	**Stopped Too Soon/ Applied Too Long**
1.Set Aircraft Attitude	UCA 1.1: The PIC does not set aircraft attitude when the trimmed aircraft state is causing the aircraft to deviate from its flight plan. (H1)	UCA 1.2: The PIC sets an incorrect aircraft attitude causing the aircraft to violate separation minimums. (H1, H2) UCA 1.3: The PIC sets an aircraft attitude that is not achievable. (H2)	UCA 1.4: The PIC changes aircraft attitude at a rate that will damage the airframe. (H3)	UCA 1.5: The PIC changes the aircraft attitude at too high or too small a magnitude when there is an obstacle nearby. (H1) UCA 1.6: The PIC changes aircraft attitude too much or too little when the aircraft is close to its flight limits. (H2)
2. Set Aircraft Power	UCA 2.1: The PIC does not adjust aircraft power when	UCA 2.2: The PIC sets the aircraft power too high causing the aircraft	UCA 2.4: The PIC changes aircraft	N/A

	there is an obstacle approaching the aircraft's position. (H1)	to exceed VNE. (H3) UCA 2.3: The PIC sets the aircraft power to a value that does not allow it to complete the selected maneuver. (H2)	power at a rate that will damage the airframe. (H3)	

Unsafe control actions for automated controllers are generated in the same way as for humans. As an example, consider the control actions from the flight control computer (FCC) to the engine controller to control the aircraft.

Table 2. FCC to Engine Controller UCAs

Control Action	**Not Providing Causes Hazard**	**Providing Causes Hazard**	**Incorrect Timing/ Incorrect Order**	**Stopped Too Soon/ Applied Too Long**
11. Set desired power.	UCA 11.1: The FCC does not request the desired power level from the engine controller when the engines are turned on. (H1, 2, 3)	UCA 11.2: The FCC requests the incorrect power level from the engine controller. (H1, 2, 3)	UCA 11.3: The FCC requests the desired power setting from the engine controller with a time delay. (H1, 2, 3)	UCA11.4: The FCC stops requesting the desired power setting before the engines shut down. (H1, 2, 3)

Control Action	Not Providing Causes Hazard	Providing Causes Hazard	Incorrect Timing/ Incorrect Order	Stopped Too Soon/ Applied Too Long
12. Provide atmospheric information	UCA 12.1: The FCC does not provide the engine controller with atmospheric information when the engines are turned on. (H1, 2, 3)	UCA 12.2: The FCC provides incorrect atmospheric information to the engine controller. (H1, 2, 3)	UCA 12.3: The FCC provides outdated atmospheric information to the engine controller. (H1, 2, 3)	N/A

Developing UCAs is quite methodical when using STPA. The control actions defined in the control model are all analyzed using the four types of unsafe control. Because the structure is so easy to follow, it is possible for to insure that all of the unsafe control actions for a system are considered as long as the model accurately represents the system. Of course at the conceptual phase, the listed control actions are abstractions of those that will exist in the mature system.

Once the UCAs are identified and listed, it is possible to compare different designs to determine which hazards overlap between designs and which are unique to a specific design. The unique hazards can be compared between designs to see if a certain design has more manageable hazards. This process can be used for trade studies in order to compare the potential safety of designs fairly quickly. More detailed comparisons can be made once causal scenarios are developed.

4.5 Identify Causal Scenarios

In STPA, causal scenarios are identified for each UCA. The resulting scenarios can be used to develop safety and cyber-security design and operational requirements to eliminate or mitigate the unsafe control actions (hazards). Figure 23 shows some of the things that can go wrong in a control loop that might appear in the causal scenarios.

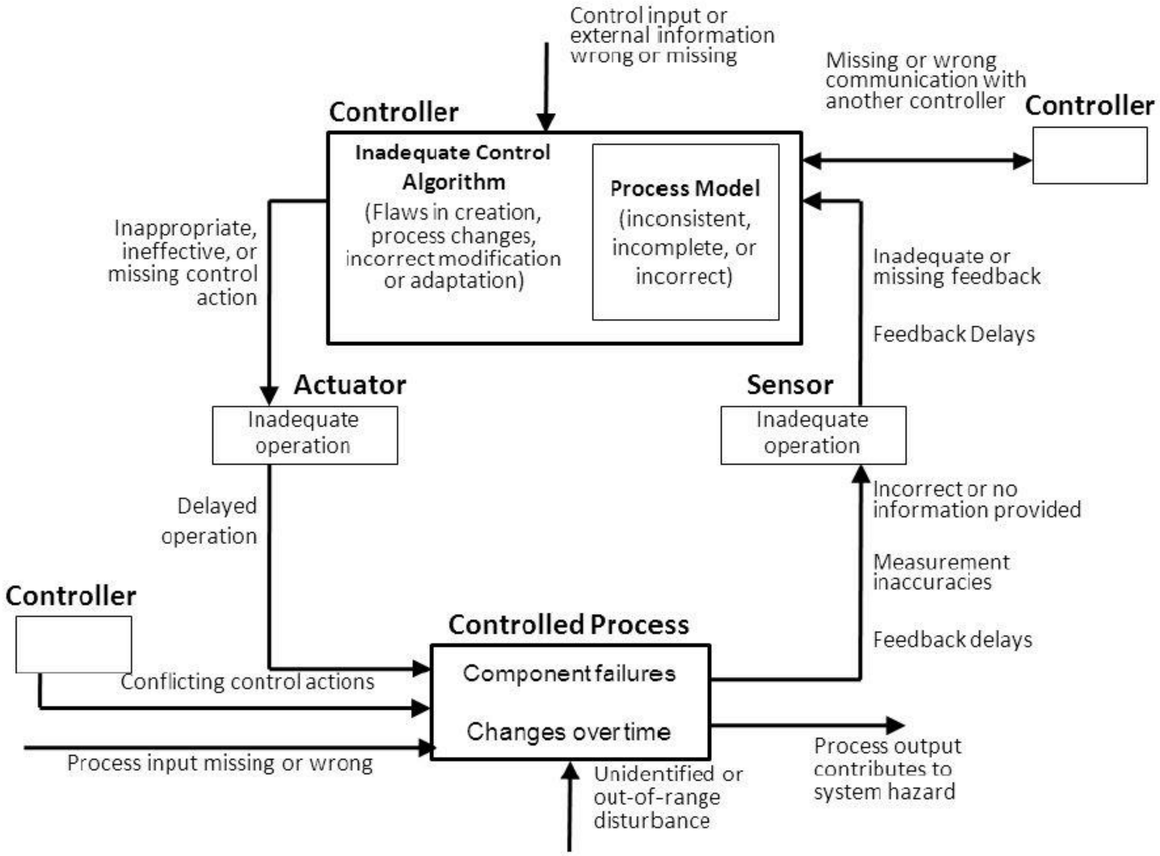

Control input or external information wrong or missing
Missing or wrong communication with another controller
Controller
Controller
Inadequate Control Algorithm
(Flaws in creation, process changes, incorrect modification or adaptation)
Process Model
(inconsistent, incomplete, or incorrect)
Inappropriate, ineffective, or missing control action
Inadequate or missing feedback
Feedback Delays
Actuator
Inadequate operation
Sensor
Inadequate operation
Delayed operation
Incorrect or no information provided
Measurement inaccuracies
Feedback delays
Controller
Controlled Process
Component failures
Changes over time
Conflicting control actions
Process input missing or wrong
Unidentified or out-of-range disturbance
Process output contributes to system hazard

Figure 23. Some control flaws that can lead to unsafe control. Note that component failures are included but control flaws other than component failures are also considered

Causal scenario generation starts with an unsafe control action. While we have not yet found an algorithm for generating causal scenarios, there are some heuristics that are useful. For example, start by identifying the flaws in the process model that could lead to the controller producing the unsafe control action. Then identify how the process model could come to have those flaws, that is, how it could have become inconsistent with the real state of the controller process. Process model flaws may result from missing or incorrect feedback from the controlled process or other aircraft subsystems, incorrect updating of the model when feedback is received, etc. For some control actions, there may also be outside input that can be missing or incorrect from external actors or other controllers. There may also be coordination problems when a process is being controlled by multiple controllers. Figure 23 shows some of the flaws in a control loop that can lead to a hazard. This classification of flaws is useful in generating the causal scenarios.

STPA does not omit or oversimplify the role of humans in systems. Using STPA as it exists today, a large variety of causes related to human factors can be identified, such as inadequate situation awareness, distraction, and mode confusion. We are working on enhancing STPA to incorporate even more sophisticated human factors concepts and provide additional guidance to engineers in generating causal scenarios. The extensions to STPA involve using a more detailed process model for human controllers that better represents what humans need to make safe decisions. We also are working with human factors experts to apply additional concepts from cognitive psychology. The general goal is to provide a communication medium that will better allow human factors experts and engineers to work together to better understand why human errors occur and how to eliminate or reduce them. This work is very new, and we are only now starting to validate it on real systems.

Note that I am no expert in aircraft design and therefore the causal scenarios are almost surely incomplete. Generation of the scenarios (performing the analysis) by aircraft design experts or at least review by experts would undoubtedly create more or different scenarios. I have shown example requirements that could be generated from the scenarios but aircraft design engineers would have to identify the best ways to eliminate or mitigate the scenarios, including improved recommendations for design and operational changes to eliminate the scenarios. The results of the STPA scenarios can be used for design of system testing. It is best that STPA be conducted by personnel with expertise in safety and the system being designed.

UCA 1.1: *The PIC does not set aircraft attitude when the trimmed aircraft state is causing the aircraft to deviate from its flight plan.* (H1)

Example Causal Scenarios for UCA 1.1a: The PIC does not set aircraft attitude when the trimmed aircraft state is causing the aircraft to deviate from its flight path because the PIC believes that the aircraft is on the desired flight path. This could occur if:

1. The PIC does not receive detailed enough feedback through the PVI to detect small deviations from the flight plan. These small deviations become greater over time and the aircraft could violate separation with an obstacle. This problem can be compounded in tethered scenarios where the PIC is responsible for multiple vehicles and cannot easily check clearances for the tethered vehicles.

2. The PIC did not program a flight plan into the aircraft systems and is unable to project a flight path for the aircraft in the airspace because of degraded conditions or a high workload. This results in disorientation and a loss of aircraft control or violation of separation standards.
3. The aircraft is incorrectly indicating that it is following the desired flight path. This could occur if the navigation system or PVI supplies inaccurate or insufficient feedback to the PIC due to degraded operation, enemy kinetic/cyber attack, or poor design.

As design decisions are made, the level of detail in the causal scenarios can be increased. An iterative process can be implemented where the general scenarios are used to refine the design, which in turn leads to more detailed scenarios that must be eliminated. One important result of this early causal analysis is the identification of the necessary feedback to eliminate or reduce unsafe control. This required feedback can then be designed into the system from the beginning.

In addition to process model flaws, the control action could be unsafe due to an overloaded pilot. Causal scenario 1.1b is an example that includes pilot overload.

Causal Scenario 1.1b The human PIC realizes that the aircraft is deviating from its flight plan but wants to make infrequent larger corrections rather than continually correcting the flight path because the pilot is overloaded. These larger corrections result in an unsafe aircraft state such as violation of separation minimums.

STPA also includes consideration of the cases where the controller does the right thing (issues a correct and safe control action) but the provided control action is not executed. These are the standard type of failure-oriented causal scenarios generated by traditional hazard analysis methods. Causal Scenario 1B shows an example where the pilot inputs a correct aircraft pitch but the system does not execute this command.

Causal Scenario 1B: The PIC sets appropriate aircraft pitch but the command is not correctly executed. This could occur if there is a hardware failure, design flaw in the control path between the PIC and the aircraft hardware systems or, in the case of cyber-security, there is an attack on the control path

1. A failure of the flight controls or their connection to the software based PVI.
2. An error in communication between the PVI and FCC.

3. An error in communication between the FCC and the flight actuators.
4. A mechanical failure of the flight actuators.
5. A delay in the control path that results in an unsafe maneuver.

Again, these scenarios may be very general before any detailed design decisions are made. At this very early stage of development, the requirements generated from the basic STPA causal scenarios will provide guidance to those making architectural and hardware design decisions. These design decisions may include focused but standard reliability and security enhancing techniques. Additional design decisions may be possible that completely eliminate the potential scenario.

The examples shown so far include only a human controller. As an example of an automated controller, consider again the case of the FCC controlling the engine controller.

UCA 11.2: The FCC requests the incorrect power level from the engine controller. (H1, 2, 3)

Causal Scenario 11.2a: The FCC requests the incorrect power level from the engine controller because it has a flawed process model of the aircraft and/or the operating environment. This could occur if:

1. The FCC is operating based on incorrect data about the environment due to incorrect or insufficient feedback and control inputs (e.g. incorrect sensor data). It passes incorrect data to the engine controller which results in the wrong power settings.
2. The FCC incorrectly believes that there is an engine fire and requests that the engine be shut down to extinguish the fire. Shutting down the engine results in insufficient power for the maneuver.
3. Data in the FCC is corrupted which results in incorrect information being passed to other aircraft systems that depend on the FCC.
4. The FCC receives an incorrect weight on wheels (WOW) indication and commands the engine controller to bring the engine to idle. This could occur if the WOW sensors malfunction or the aircraft inadvertently touches down on a surface that is not safe to land on.

5. The FCC has an incorrect engine model because the model was developed incorrectly, based on hardware/software design reused on another aircraft, or the engines are performing unexpectedly.

4.6 Derive Safety Requirements

From the causal scenarios, safety and cyber-security design recommendations or requirements can be generated to help in eliminating or mitigating the unsafe control. The requirements do not necessarily map to a specific scenario but instead are intended to prevent the scenarios leading to hazards. One requirement may address many issues and one causal scenario may drive numerous requirements.

Example requirements derived from scenario 1.1a include:

a. Navigation systems and interfaces shall allow for navigation with error less than TBD miles in manual flight modes.

b. The navigation systems and PVI shall be monitored for faults at TBD Hz to ensure that they are updating.

c. Navigation accuracy shall be confirmed through multiple independent sensors.

d. The PIC shall be alerted when the aircraft deviates from the flight plan by TBD feet.

e. The PIC shall be alerted if there is an object within TBD feet of separation minimums.

f. The aircraft shall provide the PIC with clear feedback to indicate what responsibilities the computer systems are currently taking.

g. The aircraft shall alert the pilot if there are uncommanded mode changes by the computer systems.

h. The aircraft shall monitor the PIC’s state of arousal and simplify the feedback presented if the PIC has degraded abilities and is unable to operate under a high workload.

i. The PIC shall be provided with separation information about tethered vehicles and a visualization of the entire formation relative to the flight environment.

j. A minimal flight plan shall be provided for all missions so that the aircraft can be reoriented to the flight path if the PIC becomes disoriented.

k. Data sources for informational displays shall be specified. Integrity validation shall be performed on all data sources.

l. Explicit specification of data formats shall be provided for informational displays so that the displays can be verified to correctly parse data and correctly respond to malformed data.

Safety and security requirements may be generated for the system as a whole as well as for the individual components in order to eliminate or mitigate the unsafe control actions. In addition, as mentioned elsewhere, new safety requirements can be incorporated into the requirements set and can be used to refine the safety analysis incrementally. These new requirements become part of the refinement of the design as it progresses. Because the STPA analysis starts from a control model of the entire system, in this case the aircraft, the implications of changes or additions of design detail on the aircraft as a whole (including negative impact on other parts of the system) can be identified.

Safety-guided design allows engineers to consider the hazards in the system from the very beginning and begin risk mitigation right away. By including safety in the design tradeoff analysis, design teams can more accurately assess the risks in the system and make informed design decisions. It is also possible to identify design options that have fewer or more easily mitigated hazards by applying STPA during the initial stages of the design process. Many more examples of UCAs, causal scenarios and the associated requirements are included in Appendix B.

5. **Using STPA in a Trade Study**

In order to gain a better understanding of the utility of safety-guided design, here is an example of using STPA for a tradeoff analysis. As this analysis is based on the larger system, the high level mishaps and hazards are still relevant and do not change. This example focuses on the implementation of the desired tethering capability. Two different methods of implementing tethering control will be considered. It is assumed that the lead aircraft and its tethered counterparts will travel together in a formation. It is also assumed that a single software enabled controller is primarily responsible for executing the tethering mission in each tethered aircraft.

There are numerous different formations used in aviation for different phases of flights and circumstances. Some formations make the group more defensible while others increase fuel efficiency. Transitioning through all the phases of flight also requires different formations throughout a mission. In this study, the controller that implements that command to set a formation shape will be varied. In architecture one, the human pilot in command will determine the formation shape from the lead aircraft, and the tethered aircraft will be responsible for implementing the command by maintaining their position in the specified formation. In the alternate configuration, architecture two, the tethered aircraft will determine the optimal formation shape based on the present conditions and the current phase of flight.

The formation shape and each aircraft's position will be presented to the lead PIC. In this case, the tethered aircraft will share sensor information and use the data to decide on the best formation shape to suit the conditions. Figure24 shows the general control architecture for a tethering scenario. This architecture is valid for both scenarios described above. The only thing that changes is the controller responsible for the control action 'set formation shape.'

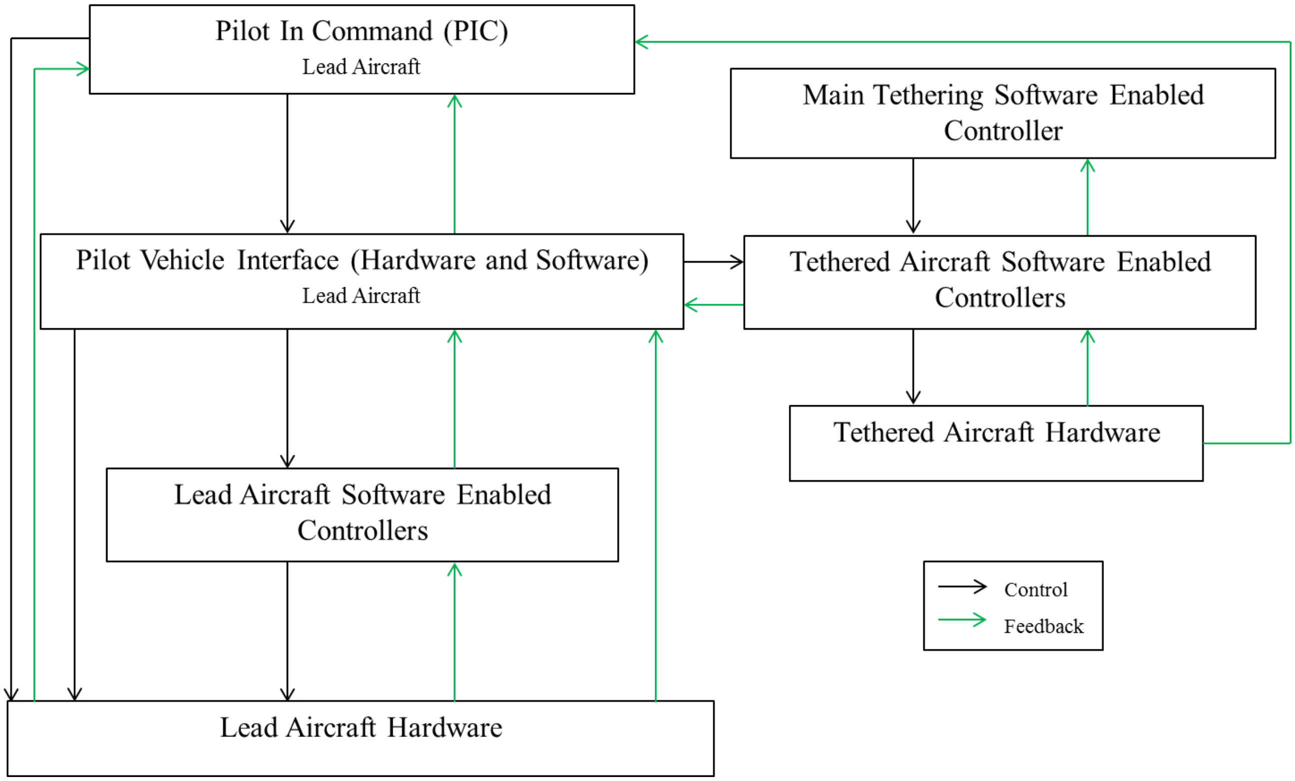

Pilot In Command (PIC)
Lead Aircraft
Main Tethering Software Enabled Controller
Pilot Vehicle Interface (Hardware and Software)
Lead Aircraft
Tethered Aircraft Software Enabled Controllers
Tethered Aircraft Hardware
Lead Aircraft Software Enabled Controllers
Control
Feedback
Lead Aircraft Hardware

Figure24. Tethering Control Structure

Figure25 shows the different placement of the control action more specifically. The main difference is the source of the "set formation shape" control action.

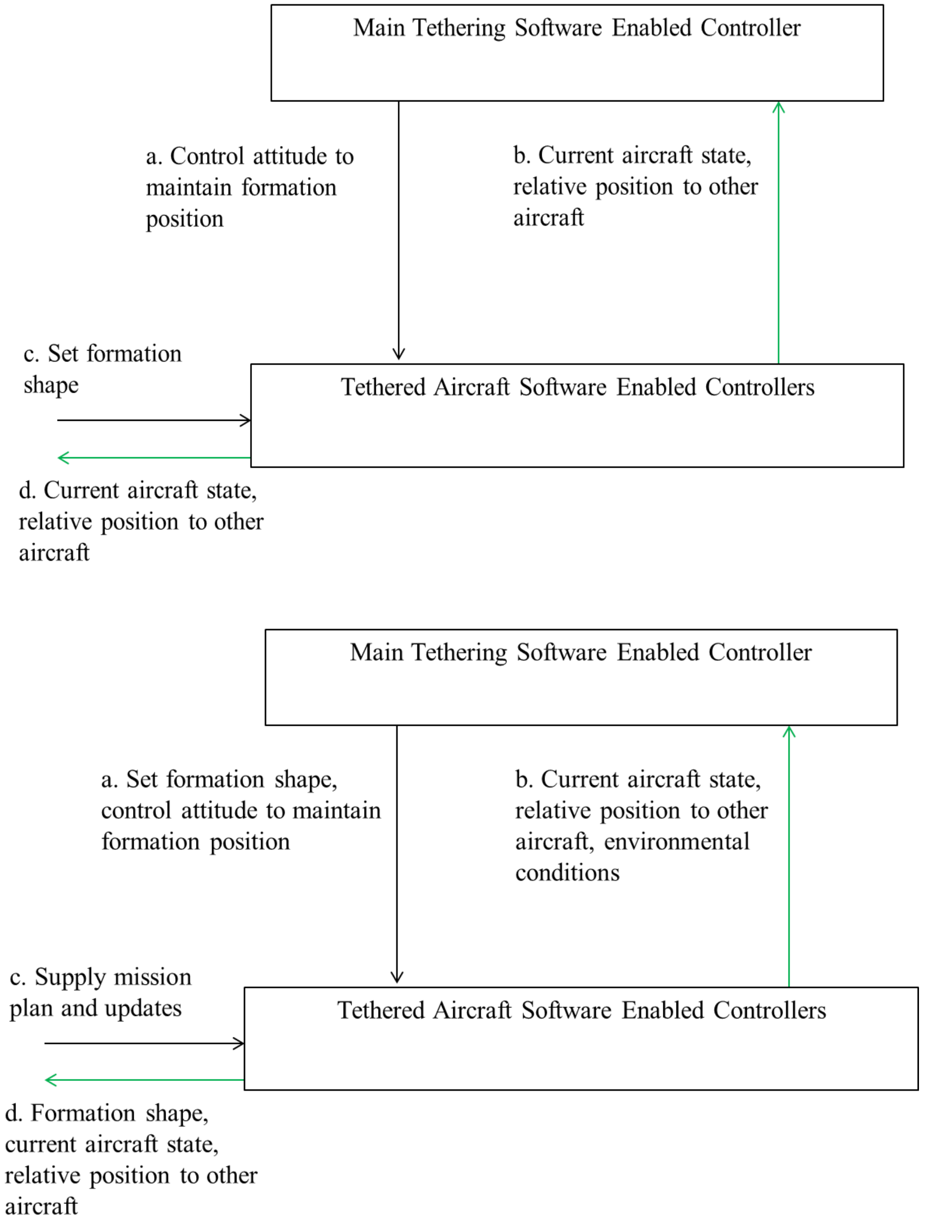

Main Tethering Software Enabled Controller
a. Control attitude to maintain formation position
b. Current aircraft state, relative position to other aircraft
c. Set formation shape
Tethered Aircraft Software Enabled Controllers
d. Current aircraft state, relative position to other aircraft
Main Tethering Software Enabled Controller
a. Set formation shape, control attitude to maintain formation position
b. Current aircraft state, relative position to other aircraft, environmental conditions
c. Supply mission plan and updates
Tethered Aircraft Software Enabled Controllers
d. Formation shape, current aircraft state, relative position to other aircraft

Figure25. Tethering Control Actions for Architecture 1 above and 2 below

This exercise will analyze both control strategies and find the associated UCAs, causal scenarios and example requirements that will help mitigate the hazards. Comparisons are made throughout the process to demonstrate how designers can utilize the information obtained from STPA. In the two control diagrams, the main difference is the allocation of safety responsibility. Architecture one allocates the responsibility for the formation shape to the lead PIC. Architecture two gives the tethered aircraft responsibility for the formation shape. This process will identify the safety factors for both assignments of responsibility, which can be used in the architectural decision making.

5.1 UCAs Comparison

Next, we examine the unsafe control actions for each architecture. Even with the same control action, the different contexts of the implementation changes the associated unsafe control actions. The tables below show how the hazards vary according to the source of the formation shape command.

I. Tethered A/C follow PIC supplied formations

Controller: Lead A/C

Controlled Process: Tethered A/C

Control Action	Not Providing Causes Hazard	Providing Causes Hazard	Incorrect Timing/ Incorrect Order	Stopped Too Soon/ Applied Too Long
1.Set Formation Shape	UCA 1.1: The lead aircraft PIC does not set a new formation shape when needed. (H1, 2)	UCA 1.2: The lead aircraft PIC sets an unsafe formation shape for the current environment. (H1, 2)	N/A	N/A

II. Tethered A/C determine best formation given environment and mission.

Controller: Main Tethering Software Enabled Controllers

Controlled Process: Tethered Aircraft Software Enabled Controllers

Control Action	Not Providing Causes Hazard	Providing Causes Hazard	Incorrect Timing/ Incorrect Order	Stopped Too Soon/ Applied Too Long

2.Set Formation Shape	UCA 2.1: The tethered A/C are unable to agree on a formation shape and none is set. (H1, 2) UCA 2.2: The tethered A/C do not provide the formation shape to the lead PIC. (H1, H2)	UCA 2.3: The tethered aircraft set an unsafe formation shape for the current environment. (H1, 2) UCA 2.4: Multiple tethered aircraft set different formation shapes in unison and maneuver into the disparate formations. (H1, H2)	UCA 2.5: The tethered A/C respond to the new formation shape at different times. (H1, 2) UCA 2.6: The tethered A/C do not have an accurate mission plan and set a formation for the incorrect phase of flight. (H1, 2) UCA 2.7: The tethered aircraft change formation shape too frequently making it difficult for the lead PIC to keep an up to date process model of the formation. (H1, H2)	N/A

As seen above, architecture two has more unsafe control actions and thus associated hazards. This is true because responsibility resides in multiple controllers. Unless one of the tethered vehicles is specifically designated to choose the formation shape, this architecture relies on the agreement of multiple entities to issue the control action.

Furthermore, the lead PIC is still ultimately responsible for the safety of the formation and thus must be informed of the decision as it affects the way that the formation is controlled as a whole. The PIC must know how the formation plans to follow his lead in order to make appropriate piloting decisions. The presence of additional hazards does not necessarily mean that architecture two is more dangerous than architecture one. It simply means that designers will have to consider additional factors during design to assure the safety of the system.

In some cases, an architecture can have a large number of hazards that can easily be mitigated. Other architectures with fewer hazards could still be more dangerous if the hazards cannot be mitigated. In this case, delegating the responsibility to set the formation shape to the tethered vehicles could decrease the workload for the PIC and thus make the mission safer as long as the implementation is carefully thought through and relevant hazards are addressed.

While identifying the unsafe control actions can give a preliminary idea of the hazards associated with a specific architecture in a trade analysis, it is beneficial to continue the analysis by identifying causal scenarios and requirements to mitigate the hazards.

5.2 Causal Scenarios

The causal scenarios and requirements that help mitigate the scenarios are listed below for each architecture. Some of the scenarios are very similar between architectures while others are unique. This information can be used in a trade analysis. Commentary on the analysis is provided in italics.

Arc Architecture One (A1):

UCA 1.1: The lead aircraft PIC does not set a new formation shape when needed. (H1, 2)

Causal Scenario 1.1a: The lead aircraft PIC does not set a new formation shape when needed because the PIC believes the current formation shape is sufficient. This could occur if:

1. The lead aircraft PIC is not able to predict future states of the formation and therefore does not know that a new formation shape is needed to avoid a conflict or unsafe flight configuration.
2. The lead aircraft PIC is task saturated and cannot generate an accurate process model of the entire tethered formation and the environment they are operating in.
3. There is not sufficient feedback from the tethered aircraft for the lead aircraft PIC to determine the best formation shape for the situation at hand.
4. There is malformed feedback from the tethered aircraft for the lead aircraft PIC to determine the best formation shape for the situation at hand. This may be in the form of incorrect position information, dropped feedback, communication with tethered aircraft has been lost, or malformed data that is not displayable by system.

Example Requirements for 1.1a:

a. The lead aircraft PIC shall be provided with feedback to predict future states of the formation. It is a difficult cognitive task to predict the future state of multiple vehicles so it is likely that predictive aids will be required.

b. Studies shall be performed to determine how pilots will respond while flying a formation with tethered aircraft. The system shall be designed to keep the workload within the PIC’s capabilities even during emergency situations.
c. The tethered aircraft shall supply feedback indicating position and velocity as well as relative position to other aircraft to the lead PIC to allow the lead PIC to make informed decisions about the formation.
d. System shall indicate to PIC current communication status between lead aircraft and tethered aircraft.
e. The system shall indicate the last known good information, and corresponding age of information to the PIC in the lead aircraft.

These scenarios highlight a shortcoming of a human controller. Because of limited attention and computational resources, humans rarely achieve the optimal solution. A computer algorithm would be better suited to continuously check and optimize the formation shape for the environment. Regardless of which controller is making the decision, it is vital that every member of the formation have a process model of the formation that matches reality or unsafe situations will arise.

<u>UCA 1.2: The lead aircraft PIC sets an unsafe formation shape for the current environment. (H1, 2]</u>

Causal Scenario 1.2a: The lead aircraft PIC sets an unsafe formation shape for the current environment because they believe the new formation shape is the best shape for the environment. This process model flaw could arise if:

1. The lead aircraft does not have an accurate model of the current environment because there are degraded conditions or aircraft sensors.
2. The lead aircraft is not able to predict future states of the formation and therefore does not know that the prescribed formation shape will be unsafe.
3. The lead aircraft receives inaccurate/partial/malformed feedback about the current environment from their aircraft, the tethered aircraft, or outside parties.

4. The lead PIC is under excessive workload and does not correctly process the available information to pick the safest formation.

Example Requirements for 1.2a:

a. Formation shapes and control modes shall be designed for degraded conditions that could lead to a hazard.
b. The lead aircraft shall be provided with sufficient feedback to predict future states of the formation. It is a difficult cognitive task to predict the future state of multiple vehicles so it is likely that predictive aids will be required.
c. The lead aircraft shall receive an alert if the aircraft sensors are not providing accurate feedback about the environment.
d. In a tethered situation, feedback from all aircraft in the formation shall be integrated and compared to increase situational awareness and allow for better error checking.

Causal Scenario 1.2b: The lead aircraft provides an unsafe formation shape for the current environment because the formation is in a dynamic environment and the lead aircraft is not able to provide adequate control for the formation.

Example Requirements for 1.2b:

a. Tethered configurations shall avoid rapidly changing environment.
b. Tethered aircraft shall have independent threat detection and terrain avoidance systems that allow them to react to environmental hazards without lead aircraft input.
c. Maintaining separation with other aircraft in the formation shall take priority over avoiding other hazards in the environment. (If tethered aircraft are ever certified to transport humans, the hazard avoidance logic shall be updated to minimize the likelihood of harm to the occupants and other aircraft in the formation.)

Degraded conditions present a challenge to all aircraft and the presence of tethered aircraft in the formation compound the challenges of operating in bad conditions. It is important that the responsible controller be able to maintain an accurate process model in all operating conditions and that sensors provide adequate information for good decisions to be made.

Causal Scenario 1B: The PIC sets a safe formation shape for the tethered formation but it is not correctly implemented or followed. This could occur if:

1. There is a failure of the flight controls or their connection to the software based PVI.
2. There is a miscommunication between the software based PVI and the mission computer which is responsible for communicating with the tethered vehicles.
3. There is a hardware failure in the communication link between the lead aircraft and the tethered vehicles.
4. There is a malfunction in one or more of the tethered vehicles that does not allow them to reach the desired formation shape.
5. There is a delay in the control path causing the new formation shape to be implemented too late.
6. Malfunction in the communications between the PIC and tethered aircraft.
7. Compromised control path drops, interferes with, or manipulates the PIC commands to the mission system and/or tethered aircraft, despite receiving feedback that a new formation was commanded.

Example Requirements for 1B:

a. The WCAAS shall alert the PIC if one of the tethered vehicles is operating in a degraded condition.
b. There shall be independent backup communication systems that can be used to maintain communication within the formation in case the primary communication channels are lost.
c. Each aircraft shall have an independent loss of link plan that corresponds to its position in the formation and allows it to exit the formation safely.
d. There shall be adequate sensors on each aircraft to allow them to safely navigate the airspace to a safe landing position without control by the lead aircraft.
e. Each tethered aircraft shall have a loss of link plan that is updated throughout the mission, which allows for the aircraft to safely land as soon as possible.
f. All aircraft shall be able to autonomously coordinate with other air traffic to avoid conflict.
g. All aircraft shall be able to find a suitable landing spot in unfamiliar areas.

h. The tethered vehicles shall send a message to the lead aircraft when they receive and act on commands. If a confirmation message isn’t received within TBD seconds, the PIC must be alerted.

Hardware failure is always possible and presents challenging circumstances. Increasing the responsibility of a PIC beyond their own aircraft increases the likelihood of a malfunction and greatly increases the workload in the case of an emergency. It is important that designers consider how emergency situations will be handled and how control actions will be implemented if the system is not performing as designed. These scenarios highlight the fact that tethered vehicles will have to act autonomously if an emergency situation arises.

Architecture Two (A2):

UCA 2.1: The tethered A/C are unable to agree on a formation shape and none is set. (H1, 2)

Causal Scenario 2.1a: The tethered A/C are unable to agree on a formation shape because they have different process models of the environment. This could occur if:

1. The tethered A/C each rely on their individual sensor information to create a model of the environment and determine the best shape for the formation.
2. The tethered A/C do not send feedback to the other A/C about formation priority rankings.
3. The feedback from the aircraft cannot be compiled into a coherent model of the formation due to missing information because of failed sensors, bad weather, or improperly calibrated instruments.

Example Requirements for 2.1a:

a. Sensor data from all aircraft in the formation shall be compiled to create a more complete model of the formation.
b. Tethered aircraft shall include in the feedback the formation priority rankings.
c. A/C shall have sensors that can determine precise position and velocity in degraded conditions.
d. There shall be backup methods of determining position and velocity and communicating state data between A/C.
e. Instruments shall be checked for proper calibration before flight.

Causal Scenario 2.1b: The tethered A/C are unable to agree on a formation shape because they are in a rapidly changing environment and the incoming data do not converge to a single solution.

<u>Example Requirements for 2.1b</u>:

a. The lead A/C PIC or one of the tethered A/C shall be able to make an overriding decision for the formation shape if there is not agreement.
b. Transient data shall be averaged to approximate the steady state atmospheric and environmental state.

Causal Scenario 2.1c: The tethered A/C are unable to agree on a formation shape because the data point to multiple formation shapes that will fulfil the needs of the A/C equally well.

<u>Example Requirements for 2.1b</u>:

a. The lead A/C PIC or one of the tethered A/C shall be able to make an overriding decision for the formation shape if there is not agreement.
b. If the tethered A/C is unable to pick a formation, the formation that reduces the workload for the PIC shall be chosen.

These scenarios present the challenge of allocating the responsibility to choose a formation shape to a group of controllers rather than a single entity. While combining sensor data and computational power can be advantageous, it becomes possible for the controllers to disagree. One way to combat this problem is to assign a lead controller among the tethered aircraft that makes the final decision.

Another drawback to allocating this decision to a computer is that computers often struggle to make choices when data lead to ambiguous conclusions. Humans can typically make a decision even if not all the data agree whereas computers may struggle if the logic does not point to a clear solution. The system must be designed to cope with ambiguous or unexpected inputs. The next scenario points to the important responsibility of keeping the lead pilot informed. At times, delegating control lessens the PICs awareness of how the system is operating and could result in confusion.

UCA 2.2: The tethered A/C do not provide the formation shape to the lead PIC. (H1, H2)

Causal Scenario 2.2a: The tethered A/C do not provide the formation shape to the lead PIC because they were not programmed to do so.

Example Requirements for 2.2:

a. The PIC shall receive feedback from the tethered A/C to include formation shape and current position so that they can create an accurate process model of the formation.

UCA 2.3: The tethered aircraft set an unsafe formation shape for the current environment. (H1, 2)

Causal Scenario 2.3a: The tethered A/C set an unsafe formation shape for the current environment because they have an incorrect model of the environment. This could occur if:

1. The sensors providing feedback about the environment are inadequate for the conditions or are malfunctioning.
2. There is a breakdown in communication in the formation and information is not shared between A/C.

3.

Example Requirements for 2.3a:

a. If the sensors do not provide adequate information to create a model of the operating environment, the tethered A/C shall precisely follow the lead aircraft in a line formation.
b. There shall be backup communication links that allow state information to be shared if the primary channels are not functioning.

Causal Scenario 2.3b: The tethered A/C set an unsafe formation shape for the current environment because the formation is in a dynamic environment and the A/C are not able to provide adequate control for the formation.

Example Requirements for 2.3b:

a. Tethered configurations shall avoid rapidly changing environments.
b. Tethered aircraft shall have independent threat detection and terrain avoidance systems that allow them to react to environmental hazards.
c. Maintaining separation with other aircraft in the formation shall take priority over avoiding other hazards in the environment. (If tethered aircraft are ever certified to transport humans, the hazard avoidance logic shall be updated to minimize the likelihood of harm to the occupants and other aircraft in the formation.)

UCA 2.4: Multiple tethered aircraft set different formation shapes in unison and maneuver into the disparate formations. (H1, H2)

Causal Scenario 2.4a: The tethered A/C set different formation shapes because they are unaware that there are other aircraft in the tethered formation. This could occur if:

1. The lead PIC does not properly initiate the tethering and the tethered A/C are not aware that there are multiple A/C being tethered.
2. There is a loss of communication between tethered A/C and it is assumed that the other A/C have exited the formation.

Example Requirements for 2.4a:

a. If A/C are flying in close proximity, they shall interrogate each other to determine the controller and mode of flight.
b. If there is a loss of communication, the A/C shall assume that they need to search for and avoid the lost A/C until its location is known.
c. A/C shall not maneuver into a new formation until they receive conformation from other A/C in formation.

UCA 2.5: The tethered A/C respond to the new formation shape at different times. (H1, H2)

Causal Scenario 2.5a: The tethered A/C respond to the new formation shape at different times because there is a delay in the communication.

Example Requirements for 2.5a:

a. Commands shall be time stamped and acknowledgements of receipt shall be sent before a new formation shape is enacted.

UCA 2.6: The tethered A/C do not have an accurate mission plan and set a formation for the incorrect phase of flight. (H1, 2)

Causal Scenario 2.6a: The tethered A/C do not have an accurate mission plan and set a formation for the incorrect phase of flight because they were given inaccurate data from the lead A/C. This could happen if:

1. The lead A/C supplied the mission plan incorrectly at the beginning of the flight.
2. The lead A/C changed the mission plan without informing the rest of the formation of the change.

Example Requirements for 2.6a:

a. The mission plan shall be displayed in a way that allows the lead PIC to periodically check for accuracy.
b. Any changes to the mission plan shall be communicated with the formation. If the lead A/C deviates from the mission plan, the PIC must be alerted to inform the tethered A/C of the new plan or return to the previous mission plan.

Implementing a safe design for tethered flight will require a high degree of coordination and communication between entities. Designers will have to put in a great deal of thought to ensure that all required feedback is exchanged among the members of the formation.

UCA 2.7: The tethered aircraft change formation shape too frequently making it difficult for the lead PIC to keep an up to date process model of the formation. (H1, H2)

Causal Scenario 2.7a: The tethered A/C change formation shape too frequently because the conditions are dynamic demanding different formations. This increases the workload of the lead PIC and demands they rapidly update their process model of the formation.

Example Requirements for 2.7:

a. If the tethered A/C have changed formation shape in the past TBD minutes, they shall maintain an acceptable but suboptimal formation shape unless safety demands that a new formation be assumed.
b. In dynamic phases of flight such as takeoff and landing, the formation shall follow the same behavior during every mission to increase the predictability for the lead PIC and other A/C. Deviations from standard procedures shall only occur to avoid violating separations or in emergency situations.

The lead PIC must not only be informed of the formation shape but the designers should design the system to operate in a way that keeps the PICs overall workload at a manageable level. Varying formation shape too much to achieve optimal efficiency may have a detrimental effect on safety if the PIC is unable to process the changes.

Causal Scenario 2B: The tethered A/C set a safe formation shape for the formation, but it is not correctly implemented or followed. This could occur if:

1. There is a miscommunication between the software enabled tethering controller and the Flight Control Computer.
2. There is a hardware failure in the communication link between the lead aircraft and the tethered vehicles.
3. There is a malfunction in one or more of the tethered vehicles that does not allow them to reach the desired formation shape.
4. There is a delay in the control path causing the new formation shape to be implemented too late.
5. There is a malfunction in the communications between the tethered aircraft.

6. Compromised control path drops, interferes with, or manipulates the commands to the mission system and/or tethered aircraft, despite receiving feedback that a new formation was commanded.

Example Requirements 2B:

a. The WCAAS shall alert the PIC if one of the tethered vehicles is operating in a degraded condition.
b. There shall be independent backup communication systems that can be used to maintain communication within the formation in case the primary communication channels are lost.
c. Each aircraft shall have an independent loss of link plan that corresponds to its position in the formation and allows it to exit the formation safely.
d. There shall be adequate sensors on each aircraft to allow them to safely navigate the airspace to a safe landing position without control by the lead aircraft.
e. Each tethered aircraft shall have a loss of link plan that is updated throughout the mission, which allows for the aircraft to safely land as soon as possible.
f. All aircraft shall be able to autonomously coordinate with other air traffic to avoid conflict.
g. All aircraft shall be able to find a suitable landing spot in unfamiliar areas.
h. The tethered vehicles shall send a message to the lead aircraft when they receive and act on commands. If a confirmation message isn’t received within TBD seconds, the PIC must be alerted.

5.3 Conclusions

This study demonstrates how to perform a safety trade analysis using STPA. Table 3 below shows how data from this analysis can be used for qualitative comparisons between the two architectures.

This analysis does not point to a clearly superior architecture on its own but illuminates potential challenges of each approach and gives designers data to use in design decisions. STPA highlights the safety related ramifications of design decisions and can be used in concert with other trade studies to drive design decisions. The results above highlight the hazards that must be addressed and identify possible strategies for mitigations.

STPA guides design by allowing designers to conduct a safety trade study without building or testing a physical system. It can also help designers decide which simulations or tests would be most profitable to perform. This analysis also highlights human factors issues, interfacing challenges, sensor and communication requirements, and many other engineering subtleties that are often overlooked in the early stages of design. A trade study of this scale takes a single engineer who is well acquainted with performing STPA about a day to complete. Further control actions could be considered with a slightly lower time cost once the basis of the analysis such as the control structure are in place.

Table 3. Comparison of Tethering Architectures

Component	**Comparison**
Lead PIC Process Model	Architecture 1 (A1) involves the PIC more and thus their process model is more likely to be updated if the formation changes. Requiring the PIC to choose the shape invests them more in the tethering activity likely increasing situational awareness. (SA)
Tethered A/C Process Models	Both architectures, A1 and A2, should have the same general process model for the tethered A/C. It is possible that requiring the tethered A/C to make piloting decisions would result in a more robust sensor system and process model as design plays out.
Lead PIC Workload	A2 would not require the lead PIC to perform as many tasks but the number of tasks assigned is not necessarily the cause of high workload. Experiments should be done to compare workload between the architectures.
Hardware	The hardware should be the same. As stated above, requiring tethered A/C to perform processing tasks could affect the hardware choices.
Software Design	Certifying tethered A/C to make piloting decisions would require

Component	Comparison
	more stringent software development. As seen in the analysis, A1 would still require the tethered A/C to make individual piloting decisions in case of an emergency.
Airspace Certification	Agencies such as the FAA and military airworthiness authorities should be consulted to determine if there would be differences in the certification processes for A1 and A2.

6. Integrating STPA into the DoD Acquisitions Process

This report has demonstrated how STPA can be applied to early concepts to create PHLs and PHAs and how to perform safety-guided design and tradeoff studies. This section aims to provide an overview of how STPA fits into the larger DoD acquisitions process. It summarizes the process from concept to retirement in order to point out where STPA can be used to help identify hazards, improve processes and ultimately reduce risk in the program. The STPA process helps engineers to see where there are shortcomings in feedback and control. These improvements are useful in a multitude of circumstances throughout the lifecycle of a program. Before diving into the acquisitions process, STPA is used to highlight the hazards for a program as a whole in order to facilitate high level planning and organizational structure and processes.

6.1 Using STPA to identify and mitigate risks to the program.

In addition to the system mishaps and hazards, every DoD project, large or small, shares the same basic high level program mishaps or losses. Program managers and all of the teams that interact to produce a product want to avoid common losses while building the best possible solution. The high level program mishaps are listed below.

Program Mishap1: Project cancellation
Program Mishap 2: Financial loss (Project goes over budget)
Program Mishap 3: Loss of time (Project falls behind schedule)

Every member of a team could tell you that they would prefer to be on time and below budget. It is also self-evident that any project manager would like to see their program to completion and solve the problem they were tasked with. If these mishaps are well known and every member of the team is working to avoid them, why do so many projects encounter at least one of these mishaps during their lifecycle? The answer is often simple: the problems that DoD workers are tasked with solving are difficult complex problems that require coordination, new technologies, and frequent vetting, and must be solved in the rigidly regulated framework set up to ensure that tax dollars are spent responsibly. While STPA cannot turn this complex process into a simple one, it can help illuminate the path of least resistance.

First, STPA can help provide a common process model of the problem that must be solved for everyone in the program. STPA control structures can be used to illustrate the technical and organizational architectures of the project in a way that is concrete and informative for everyone involved. Making the assumed architectures explicit can help members of the team better understand their role in the project as a whole and illuminates shortcomings that may not be easily identified otherwise. This has already been demonstrated in Figure 19. Figure 26 demonstrates an example model of the organizational control structure behind a typical DoD program.

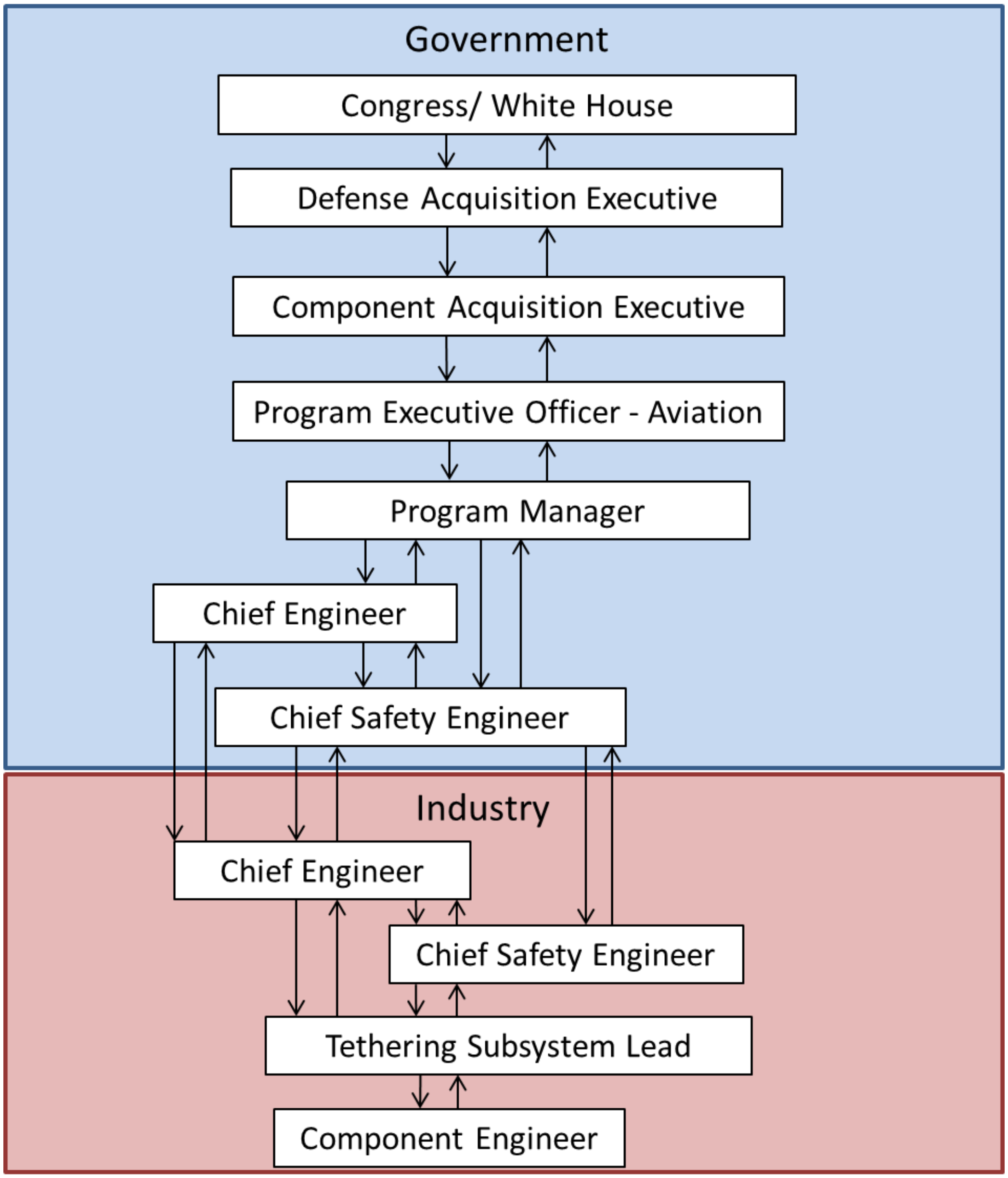


Figure 26. Control of a DoD acquisitions program.

The control structure shown is a gross simplification for a large program but nonetheless illustrates flow of responsibility and information in a program. Safety related responsibilities can be assigned to each person shown in the control structure. It is important that each controller has the proper authority and information to carry out their assigned safety related responsibilities. This seems obvious but it can often be difficult to confirm if the control relationships and responsibilities are not made explicit. For example, it could be the tethering subsystem lead's responsibility to assess the safety of the tethering capability. If they find an issue that cannot be resolved without changing the high level system design, they need access to the chief engineer on the industry side and possibly even the government engineers and program manager. If there is not a clear path of feedback and communication, it becomes difficult or impossible for lower level engineers, who are ultimately the system experts, to give meaningful input to shape the program as a whole.

In addition to explicitly illustrating the control within a program and assigning safety related responsibilities, an STPA analysis can be fully completed on the organization to help eliminate hazards in the acquisitions process. As an example, we will examine a single control action from the Program manager to the chief engineer in industry. The control action is “modify subsystem requirement.” The UCAs are shown below.

Controller: Program Manager

Controlled Process: Chief Engineer (Industry)

Control Action	**Not Providing Causes Hazard**	**Providing Causes Hazard**	**Incorrect Timing/ Incorrect Order**	**Stopped Too Soon/ Applied Too Long**
1.Modifty Subsystem Requirement	UCA 1.1: The program manager does not modify a subsystem requirement when analysis has shown the requirement leads to hazards. UCA 1.2: The program manager does not modify a subsystem requirement when the stakeholders have changed their priorities.	UCA 1.3: The program manager modifies a subsystem hazard without considering the effects of the change on the safety of the system.	UCA 1.4: The program manager modifies a subsystem requirement after the subsystem has been incorporated into the system.	N/A

Some of the UCAs relate to the system hazards as they affect the safety of the resulting system while others relate to the program mishaps. Most of the UCAs that affect the safety of the resulting system will eventually result in a program mishap because they will lead to redesign or an eventual mishap during operations. Defining control actions can be difficult for operations because design is not always a well-scripted process. Control actions can be generalized or related to specific program milestones. This example is continued with the definition of causal scenarios and example requirements. UCA 1.4 is the example UCA.

<u>UCA 1.4: The program manager modifies a subsystem requirement after the subsystem has been incorporated into the system.</u>

Causal Scenario 1.4a: The program manager modifies a subsystem requirement after the subsystem has been incorporated into the system because they have an inaccurate process model of the program and its progress. This could occur if:

1. There is not a continuous stream of feedback indicating program progress.
2. The program manager does not understand the feedback that is being provided.
3. The industry team is supplying misleading feedback about their progress to mask performance issues or other problems.
4. There is a delay in the communication between industry and the program manager.
5. The program manager is only receiving partial feedback.

<u>Example Requirements:</u>

a. Feedback shall be given electronically so that it is continuously updated.
b. Feedback shall be provided in an understandable format (specified by program manager).
c. Audits shall be implemented to ensure that the feedback to the government reflects the actual state of the project.
d. Feedback to lower level managers shall be made available to the program manager for review.

This information can be used to help design efficient operations and make sure that the proper communication channels are established within the organization. This quick analysis shows the format of an operations analysis but only scratches the surface of the value of this sort of approach. A full analysis is outside of the scope of this work. STPA can be used to help design an operation plan or to correct deficient operations.

In the case of operations, many of the actuators and controlled processes that make up the organization are people. Rather than a mechanical or software based component, we rely on other people to carry out the control actions within an organization. People are not as predictable or reliable as machines often are, which makes designing an organization different than designing a machine. People make decisions and allocate their efforts based on a complex weighing of opportunity costs and preferences. Where machines need strict guidelines and design criteria, people need good leaders and ample instruction. It is important for managers to realize that an organization with an excellent structure can still fall short if the people in it are not trained, motivated, and focused on the task. The execution of the organizational plan is just as important as the design. STPA is an excellent tool for designing an organizational structure but much work is required to turn a plan into a functioning organization. Other methods must be sought out in order to accomplish the implementation. A well designed organization coupled with effective leadership and management can reduce the risk of a program mishap.

6.2. A Review of Safety Guided Design

STPA can be used throughout the acquisitions process. As stated above, STPA can be used to setup the process itself. Previous sections have demonstrated how to integrate STPA into the design process through concept formation, tradeoff analysis or analysis of alternatives, and into detailed design.

Concept Formation:

As demonstrated, STPA can be used to illuminate hazards during concept development. STPA gives teams a structured approach to identifying hazards that is applicable even at the earliest stages of design. The use of STPA in concept formation helps shape a concept with minimal hazards and can have a large impact on the cost of implementing a safety program throughout the life of a program.

Tradeoff Analysis and Analysis of Alternatives

Section five demonstrates how to use STPA to create a safety tradeoff analysis. STPA is a tool that can be used to gather safety information for use in tradeoff analysis. This allows for safety to be considered along with other performance characteristics.

Detailed Design

Safety-guided design as demonstrated in section four is to be used from concept formation until productions. As a design becomes more detailed, the control structure becomes more specific. Rather than using hypothetical control actions, specific control actions are analyzed. Once causal scenarios are formed the resulting recommendations can be considered and implemented into the physical design. STPA works with many design approaches because of its flexibility. STPA is most easily integrated into the traditional iterative design approach based on careful systems engineering. Other, potentially more rapid design approaches can also use STPA as one method of rapid prototyping. An STPA control structure helps engineers see system behaviors and characteristics without building the design. An understanding of the STPA analysis process can also help engineers diagnose problems in designs as they can analyze feedback and control issues more easily once they are familiar with the process. Performing an STPA analysis that spans from concept formation to product fielding also creates clear traceability for safety throughout the design process. STPA provides clear rationale for each requirement that is created that can be traced to the highest level hazard. [21] Having such thorough documentation for safety related design choices makes it easy for engineers to evaluate later decisions based on the work that has been completed. This sort of documentation helps eliminate the replication of effort and can save a great deal of time and help engineers avoid changing decisions that were based on sound analysis. STPA essentially provides a coherent safety story throughout the lifecycle of a system.

STPA can also be used in concert with other techniques. Many engineers have gotten comfortable with other safety and reliability analysis techniques such as fault trees. STPA is flexible enough to work alongside other analysis methods. If desired, a team could use traditional safety analysis techniques where they are best suited and use STPA to find the missed scenarios. Although this may be less efficient than using only STPA, it allows for experienced engineers to keep using the tools they are comfortable with and have developed an expertise in while still leveraging the power of STPA to find hazards that are not the result of component failure. For many, this hybrid technique could be the most efficient and effective.

6.3 Using STPA for Developmental Testing

For his PhD work, Dan Montes developed a method to apply STPA during developmental testing. [22] He detailed a method for developing a test plan using STPA and integrating the test safety effort into the larger systems safety effort for the entire project. I will summarize his work in this section to highlight the benefits of using STPA for developmental testing.

Developmental test is an inherently dangerous stage of development and safety is always a priority. STPA helps to address some weaknesses in traditional testing approach.

- There is inconsistent expert knowledge at any given test-safety review board.
- There may be minimal expertise in new technologies (e.g., software, autonomy).
- The test-safety planning process does not use common visual aids in its documentation.
- Test engineers do not have a consistent method of tracing undesirable behavior or potential design flaws to effects on the system within the context of field use; this especially affects human-engineering experts, who cannot ignore the relationship between operating philosophy and system design.
- Problem reports tend to be reductionist (e.g., manufacturing error, component defects) and do not consistently explain system impacts through anything but written narratives. [22]

STPA is a structured process that helps to illuminate potential gaps in safety that would not be found in a less structured process. This can help find problems even if there is limited expertise in a certain area. The structure of the process also helps reduce the effects of inconsistent expertise. Unlike other methods, STPA provides visual control structures that help facilitate discussion and improve the mental model of the personnel designing the safety plan. STPA also helps provide consistent traceability of hazards throughout an entire program. Since STPA can be used during every stage of design and into operations, it provides a coherent and continuous safety story that is not often available for a project. Lastly, STPA highlights problems in their entirety, noting contributing factors that can lead to a mishap. This allows designers to address the entirety of the problem and often provides flexibility by creating multiple solutions.

In order to demonstrate how to apply STPA during developmental test, an example from Montes will be used. The system under investigation is similar to the tethering capability that is being investigated in this analysis. The Air Force was testing a system that enabled a lead pilot to control another aircraft that was programmed to fly as its wingman. The wingman was programmed to fly in different formation positions as programmed by the lead pilot. The pilot could command position changes through push button controls or by performing a predetermined acute maneuver such as rocking his wings. A data link and sensing pod allowed the aircraft to remain in contact. In order to improve the safety of the system during test, the wingman aircraft had two pilots on board to monitor the system and intervene if necessary.

The test safety analysis considered six mishaps that are listed below. [22]

M1: Ground personnel are killed or injured
M2: Ground assets are damaged or destroyed
M3: Flight personnel are killed or injured
M4: Flight assets are damaged or destroyed
M5: Asset enters prohibited airspace or range
M6: Test data are lost or destroyed

These mishaps expound upon the mishaps presented in section four to address possible losses that are unique to the test environment. The testing process introduces a new environment for the system and involves different support assets than the operational context for which the system was designed. The high level hazards for this analysis were not presented.

Developmental test is a unique stage of design in which the system is modified to help maintain safety during the testing process. In order to limit variability in conditions, which could lead to a mishap or confound the test results, the test environment is sanitized. During the design and build stage, the system is represented as it will be fielded and the field environment is considered. During testing, it is important to model the system in the test environment. This difference demands that a new control structure be made to accurately reflect the system during test (SDT). Figure 27 is Montes' control structure for the test. The figure has been color coded to help clarify how the SDT varies from the fielded system. All of the components in orange are unique to the SDT. The purple box marks the item being evaluated (IBE) by the test, in this case the autonomously piloting function. This diagram contains slightly more detail than those presented in section five because this diagram represents a mature system that has been built and is being tested whereas section five evaluated concepts that have yet to be designed. The control actions and feedback associated with the diagram are explained in Table 4. Again, the feedback and control actions are more specific and well defined because of the maturity of the system.

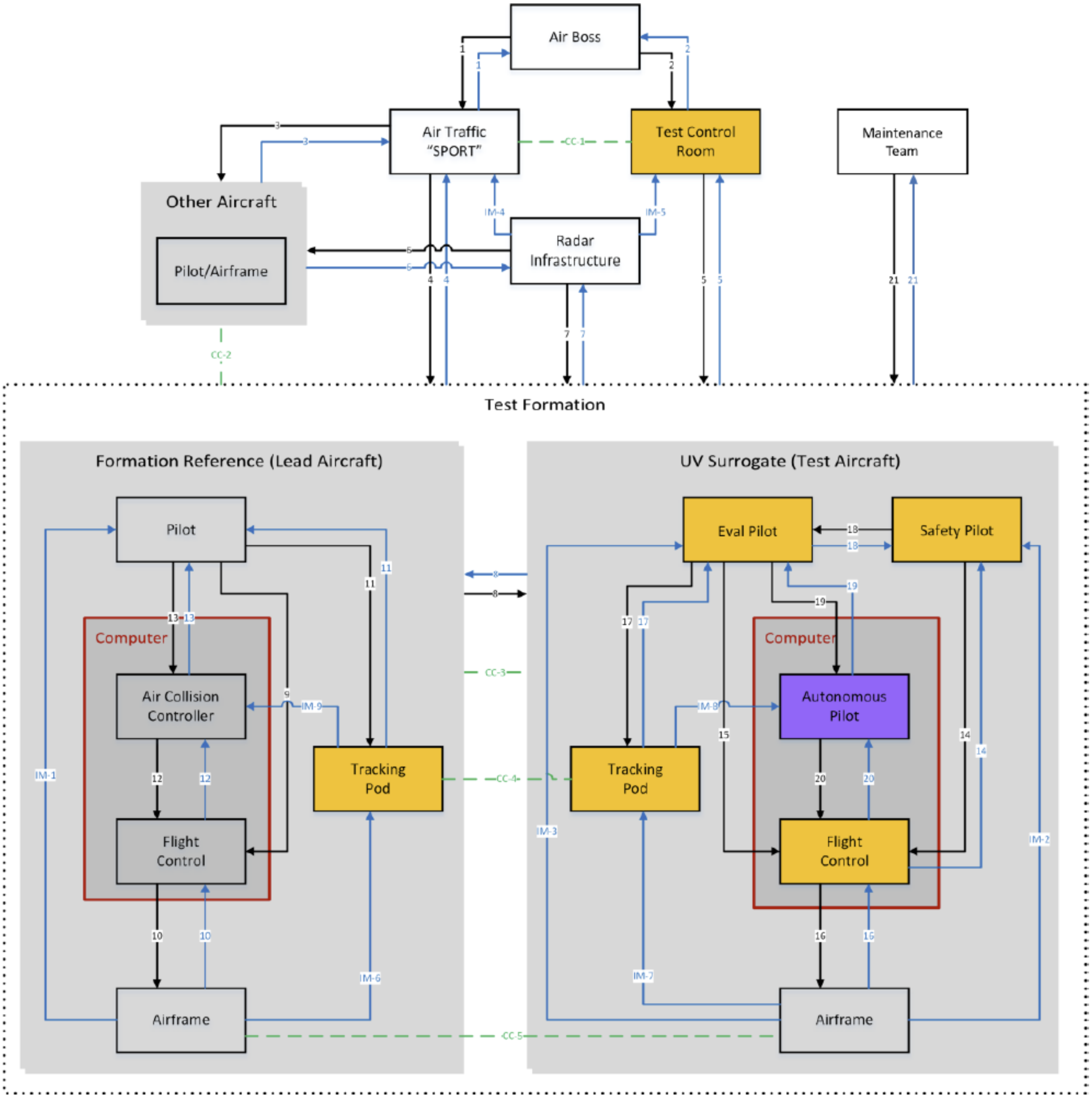
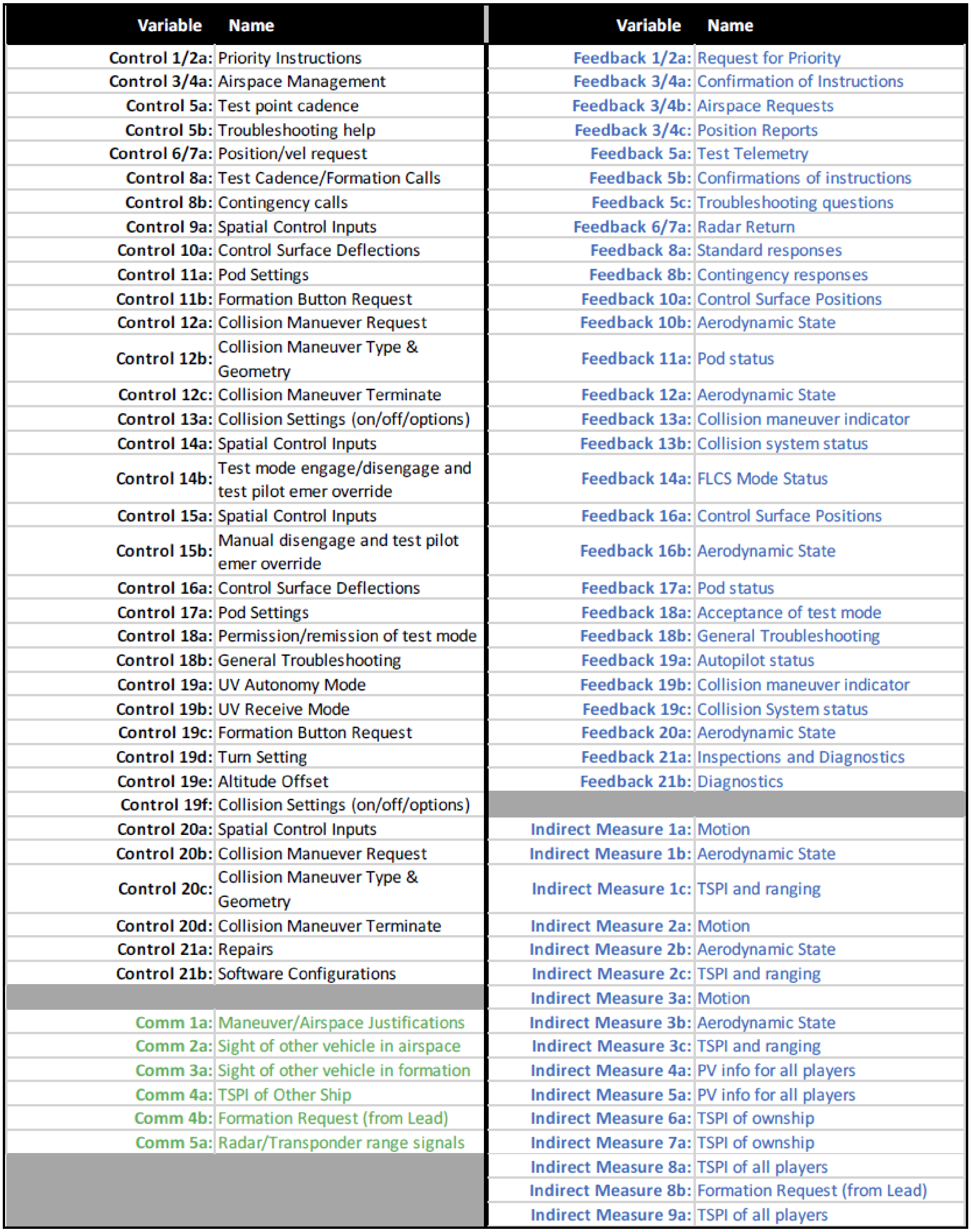


Figure 27. Autonomous Wingman Safety Control Structure. [22]

Table 4. Autonomous Wingman Variable Reference [22]



Variable	Name	Variable	Name
Control 1/2a:	Priority Instructions	Feedback 1/2a:	Request for Priority
Control 3/4a:	Airspace Management	Feedback 3/4a:	Confirmation of Instructions
Control 5a:	Test point cadence	Feedback 3/4b:	Airspace Requests
Control 5b:	Troubleshooting help	Feedback 3/4c:	Position Reports
Control 6/7a:	Position/vel request	Feedback 5a:	Test Telemetry
Control 8a:	Test Cadence/Formation Calls	Feedback 5b:	Confirmations of instructions
Control 8b:	Contingency calls	Feedback 5c:	Troubleshooting questions
Control 9a:	Spatial Control Inputs	Feedback 6/7a:	Radar Return
Control 10a:	Control Surface Deflections	Feedback 8a:	Standard responses
Control 11a:	Pod Settings	Feedback 8b:	Contingency responses
Control 11b:	Formation Button Request	Feedback 10a:	Control Surface Positions
Control 12a:	Collision Manuever Request	Feedback 10b:	Aerodynamic State
Control 12b:	Collision Maneuver Type & Geometry	Feedback 11a:	Pod status
Control 12c:	Collision Maneuver Terminate	Feedback 12a:	Aerodynamic State
Control 13a:	Collision Settings (on/off/options)	Feedback 13a:	Collision maneuver indicator
Control 14a:	Spatial Control Inputs	Feedback 13b:	Collision system status
Control 14b:	Test mode engage/disengage and test pilot emer override	Feedback 14a:	FLCS Mode Status
Control 15a:	Spatial Control Inputs	Feedback 16a:	Control Surface Positions
Control 15b:	Manual disengage and test pilot emer override	Feedback 16b:	Aerodynamic State
Control 16a:	Control Surface Deflections	Feedback 17a:	Pod status
Control 17a:	Pod Settings	Feedback 18a:	Acceptance of test mode
Control 18a:	Permission/remission of test mode	Feedback 18b:	General Troubleshooting
Control 18b:	General Troubleshooting	Feedback 19a:	Autopilot status
Control 19a:	UV Autonomy Mode	Feedback 19b:	Collision maneuver indicator
Control 19b:	UV Receive Mode	Feedback 19c:	Collision System status
Control 19c:	Formation Button Request	Feedback 20a:	Aerodynamic State
Control 19d:	Turn Setting	Feedback 21a:	Inspections and Diagnostics
Control 19e:	Altitude Offset	Feedback 21b:	Diagnostics
Control 19f:	Collision Settings (on/off/options)		
Control 20a:	Spatial Control Inputs	Indirect Measure 1a:	Motion
Control 20b:	Collision Manuever Request	Indirect Measure 1b:	Aerodynamic State
Control 20c:	Collision Maneuver Type & Geometry	Indirect Measure 1c:	TSPI and ranging
Control 20d:	Collision Maneuver Terminate	Indirect Measure 2a:	Motion
Control 21a:	Repairs	Indirect Measure 2b:	Aerodynamic State
Control 21b:	Software Configurations	Indirect Measure 2c:	TSPI and ranging
		Indirect Measure 3a:	Motion
Comm 1a:	Maneuver/Airspace Justifications	Indirect Measure 3b:	Aerodynamic State
Comm 2a:	Sight of other vehicle in airspace	Indirect Measure 3c:	TSPI and ranging
Comm 3a:	Sight of other vehicle in formation	Indirect Measure 4a:	PV info for all players
Comm 4a:	TSPI of Other Ship	Indirect Measure 5a:	PV info for all players
Comm 4b:	Formation Request (from Lead)	Indirect Measure 6a:	TSPI of ownship
Comm 5a:	Radar/Transponder range signals	Indirect Measure 7a:	TSPI of ownship
		Indirect Measure 8a:	TSPI of all players
		Indirect Measure 8b:	Formation Request (from Lead)
		Indirect Measure 9a:	TSPI of all players

After a new control structure has been made for the SDT, the standard STPA process of defining UCAs and causal scenarios is completed. Compared to typical flight test safety, this is a time consuming step but it illuminates a multitude of hazards that might otherwise be missed. During system design, safety requirements are generated from the STPA results. During test, the STPA results are used to define minimizing procedures, corrective actions and recovery actions. Minimizing procedures are actions that can be taken to help mitigate or eliminate hazards before or during test missions. They aim to reduce the likelihood of a hazard occurring. Corrective actions are actions that can help reduce the likelihood that a hazard will lead to a mishap once it has occurred. Corrective actions are often implemented during a test if a hazard is realized. Recovery actions are controls that attempt to lessen the severity of a mishap if it occurs. Together, these products are compiled to create a safety plan that will help minimize the risks during testing. Montes details how to specifically compile a flight test safety plan in his Thesis.

In order to test his approach, Montes did a comparative study with test pilots and flight test engineers at the U.S. Air Force Test Pilot School (TPS) against the traditional approach. Montes presented the evaluators with a traditional plan made by TPS students and with his STPA based plan and asked them to compare which plan was more intelligible, informative, and implementable. He evaluated the subjects using a survey with multiple choice and short answer questions. His experiment demonstrated a statistically significant result that favored the STPA approach.

6.4. Operations and Leading Indicators

All of the design work is done and the system has been produced and tested, now it’s time for the system to accomplish its mission. Operations is the goal and focus of every acquisitions process and the time for a safety program to be put to test. Just like the engines of an aircraft must be maintained for thrust to be dependable, it is necessary to continue the safety effort throughout operations in order for the design work to pay off and to avoid mishaps. STPA continues to be useful in operations for any redesign and most significantly by helping setup and design the safety management process. Leveson outlines how to use STPA for operations in “A Systems Approach to Risk Management Through Leading Safety Indicators.” This section will summarize Leveson’s work. For a thorough understanding of the use of leading indicators see Leveson’s work as well as Ball’s work on leading indicators. [23], [24] During operations, it typically becomes clear that the assumptions made during design were not entirely accurate. The system is fielded and challenged in unexpected ways, the components interact with each other and the environment, and the operators choose to control the system in way that were not anticipated. This is nearly unavoidable but it means that the assumptions made during design must be checked and any assumptions that are not being met must be investigated. One way to monitor and control the safety of a system as it drifts from its ideal designed state during operations is through the use of leading indicators. Leading indicators are warning signs that can be used to monitor safety-critical processes and detect when a safety related assumption is broken or challenged. Monitoring leading indicators can prevent mishaps if the leading indicators are carefully selected [23].

> Underlying and justifying the use of leading indicators is a belief that most major accidents do not result simply from a unique set of proximal, physical events but from the migration of the organization to a state of heightened risk over time as safeguards and controls are relaxed due to conflicting goals and tradeoffs [25]. If this belief is correct, there should be ways to detect evidence of this migration and intervene before a loss occurs [23].

A well-chosen set of leading indicators is a way to monitor the migration away from a safe state and correct deviators when they occur.

Every engineering decision is based on a set of assumptions from the very basic scientific assumptions to more specific assumptions about maintenance, operating procedures, and environment. If any of the assumptions are violated then the system will not behave as designed. Leading indicators are identified to monitor the validity of key safety assumptions.

The first step to identifying leading indicators starts at the very beginning of design. Engineers must document the assumptions they are making throughout the design process and compile them in a database that is organized and accessible by others on the project. An example assumption would be that the component engineer designing the tethering data link assumes that the hardware will be kept between -20 C and 150 C. If the service decides to utilize the aircraft in arctic conditions, this assumption would likely be violated and a mishap could occur. By monitoring this assumption, the safety team would know that the hardware would have to be updated to enable the aircraft to perform in arctic conditions. In addition to allowing for the creation of leading indicators, tracking assumptions helps eliminate rework in design and allows engineers to understand why and how earlier design decisions were made. STPA causal scenarios are the basis behind many of the safety related design assumptions and decisions. A project in which design assumptions are catalogued and available will operate more smoothly and result in a better end product than one in which they are not shared.

The causes of accidents can typically be traced back to deficiencies in three different areas: development and implementation, operations, and management. Problems arise when there was inadequate hazard analysis done during development and implementation. This includes missing hazards and incorrectly assuming that hazards were unlikely or sufficiently mitigated. Shortcomings in operations result when the controls that were designed to help the system operate safety are not properly implemented or degrade over time. Accidents can also arise if the safety management system design is flawed or if it does not operate as designed. In order to successfully monitor each of these areas, it is important to choose measurable leading indicators in each realm.

Picking leading indicators to monitor is an important step in the safety management process. A well-managed design process will result in an STPA report with a list of hazards and safety requirements as well as assumptions associated with each aspect of the design. Now the safety-critical assumptions must be identified. Leveson created a list of guidelines for finding safety critical assumptions. Safety critical assumptions generally involve: [23]

1. Assumptions about the system hazards and the paths to (causes of) hazards. New hazards may arise or assumptions underlying the causal analysis of existing hazards may change.

2. Assumptions about the effectiveness of the controls, that is, the shaping and hedging actions, used to reduce or manage hazards. For example, the flare tower in a chemical plant may be sufficient to handle the maximum amount of gas released when the plant is designed, but changes in the plant or even new information about the hazards may invalidate these assumptions over time.

3. Assumptions about how the system will be operated and the environment (context) in which it will operate. For example, assumptions that the controls will be operating as assumed by the

designers (e.g., refrigeration units would control the reactivity of the MIC at Bhopal). Assumptions about human behavior are particularly vulnerable as humans tend to adapt their behavior over time.

4. Assumptions about the development environment and processes

5. Assumptions about the organizational and societal safety control structure during operations, i.e., that it is working as designed, the design was adequate to ensure the system safety requirements are enforced, and the system controllers are fulfilling their safety responsibilities and operating as designed. For example, accident investigations often uncover the fact that some feedback and communication channels are broken or degraded and are not operating as assumed. Such assumptions include those about the state of the safety culture, for example, that the organizational safety policy is being followed.

6. Assumptions about vulnerability or severity in risk assessment that may change over time and thus require a redesign of the risk management and leading indicators system itself.

The first three assumptions are related to technical aspects of the system while the last three are related to the organization and management.

Traditional safety analyses provide probabilities of failure to components and controllers in a system. As discussed earlier, these probabilities are not always reliable. An alternate way to decide which processes propose the most risk is by considering vulnerability [23]. Unlike likelihood, vulnerability doesn’t have a specific value. To determine vulnerability an assumption is evaluated to determine if it could possibly become invalid throughout the life of a system. If an assumption could become invalid during the life of a system, the related process or component is deemed to be vulnerable and should be monitored. By evaluating risks in a binary manner, using vulnerability eliminates the practice of extrapolating probabilities of failure. Each assumption should be evaluated and marked as vulnerable or invulnerable. Vulnerability could potentially change throughout the system life so the assumptions should be periodically reevaluated as the system gains fielded experience.

Engineering assumptions are typically related to component design. Looking at the STPA analysis related to each component can help decide which aspects of the component and its use in the system are safety-critical. Some of the assumptions are innately satisfied by the design of the system and will be satisfied regardless of changing environment or operations. Other technical assumptions are validated by operating in a specific manner. Such vulnerable assumptions should be checked against the identified hazards to see if violating them could lead to a hazard. If an assumption is vulnerable and the violation of the assumption will lead to a hazard, a leading indicator should be established to monitor the assumption during operations.

Checking the effectiveness of controls starts during the testing phase of development and extends into operations. Safety controls should be continuously evaluated and updated throughout the lifecycle of a program. By identifying the underlying assumptions of the controls, leading indicators can be used to help monitor the effectiveness of controls in the field.

Other assumptions are imposed on the system by requirements or environmental constraints. The DoD often expands the operating environment of a system throughout its lifecycle if it proves effective. Before approving a system to operate in a new environment, the design assumptions must be checked with the new environmental conditions to determine if hazards will arise.

Operational assumptions are equally important to safety. One important activity that must be accomplished before operations can commence is the assigning of safety related responsibilities including the responsibility to monitor all of the leading indicators. Before a program can be fielded, somebody within the organization must be given and take responsibility for control of all the necessary functions within the system as well as checking to ensure that the system is properly sustained and managed. These responsibilities should be audited throughout the life of the system to make sure that they are properly enacted. When assigning the safety related responsibilities, it is important to ensure that team members do not have conflicting responsibilities and that any shared responsibilities have feedback paths to ensure that conflicting directives are not given.

One last, but very important assumption must be monitored. Every operation has a safety culture that is variable depending on the leadership and state of the mission. It is assumed that the safety culture is accepted and underlies decision making [23]. If this is not true, a system will inevitably migrate to a high level of risk and mishaps will become likely. It is crucial to have management that value safety and ensure that the culture of the organization is one that works to maintain safe operations despite other pressures.

Once leading indicators have been chosen, they must be managed in a way that prevents mishaps. Each leading indicator should be specified along with:

- Associated assumption(s)
- How it will be checked
- When it will be checked
- The actions to take of the indicator is true (the assumption is violated) [23]

In order to assist in the implementation of leading indicators, there are a number of tools that can be used. First shaping actions are taken to prevent hazards. Shaping actions are integrated into the design or operations plan to help control the migration to states of higher risk. An example of a physical shaping action would be a two-step process for a task such as ejection or weapons launch. Hedging or contingency actions are those that prepare for the possibility that an assumption will fail. Hedging actions are reactions to the worst case scenario, evacuation plans, fire suppressants near fuel lines and other controls that are put in place in case a hazard is realized. These two actions are used to help systems avoid and react to hazards. They are the engineering tools used to take action on leading indicators.

There are also tools that help with operations. Signposts are markers that help engineers know when a review is needed. They are events that would likely lead to a change in the system and thus demand a review. An example of a signpost is moving the system to a new operating environment. When this occurs, a review of the safety management system is needed and new shaping or hedging actions may be needed. Lastly, assumptions checking involves managers monitoring the system and specifically checking the safety assumptions periodically. At times, there may not be an obvious change that triggers a change in the system but rather a slow change may occur that causes assumptions to become invalid. It is important to methodically monitor assumption to assure that any migration towards a higher state of risk is identified.

The process of identifying and monitoring leading indicators helps manage and track risks in a system. STPA helps identify safety related assumptions and gives guidance for appropriate shaping or hedging actions. As a program moves to operations, the hazards become real. Any holes in the safety management process must be identified and corrected to avoid a mishap. Leading indicators can help managers make the correct decisions and take action to maintain an effective safety management program.

7. Conclusions

Traditional hazard analysis techniques are unable to cope with the increasing complexity of modern software enabled systems. This demands a new analysis technique that can better consider hazards that are based not on individual component failures but on systems theory. Systems-Theoretic Process Analysis is based on systems theory and the idea that safety is an emergent property. STPA examines safety as a control problem in order to identify and offer mitigation for hazards within the system. It is fitting to have a safety analysis technique built on the same theory that is used to design the systems engineering process guiding the project itself.

This effort has identified and demonstrated how STPA can be applied throughout the lifecycle of a program based on the DoD acquisitions framework. STPA can be used as the primary hazard analysis tool during concept development, analysis of alternatives, detailed design, integration and test, production and on into operations. STPA is a flexible and powerful approach that has been tested and utilized extensively in a variety of industries and applications.

This work also provided additional insight and guidance into using STPA and Safety-Guided Design for a safety trade study. Performing safety trade studies allows for safety to be considered with other system properties during concept development and design. The ability to better consider safety during the early stages of design has wide ranging benefits including better safety outcomes during the entire program and reduced costs due to safety related rework in the later stages of design.

When considered with the existing body of work on STAMP and STPA, this thesis gives compelling evidence that STPA can improve the safety management process for Department of Defense programs. The techniques described in this report guide DoD engineers to learn and apply STPA throughout the lifecycle of a program. In addition to this report and related works about STPA, it would be beneficial for a STPA handbook to be created for use by acquisitions teams and DoD contractors. A thorough handbook written expressly for the DoD that includes references to MIL-STD 882 would greatly assist in the implementation of STPA to defense programs. Additional guidance for developing causal scenarios would also prove valuable for new STPA users.

List of Acronyms:

A/C	Aircraft
ATC	Air Traffic Controller
DAL	Design Assurance Level
DoD	Department of Defense
DVE	Degraded Visual Environment
ETA	Event Tree Analysis
FCC	Flight Control Computer
FCS	Flight Control System
FHA	Functional Hazard Analysis
FMEA	Failure Mode and Effects Analysis
FMECA	Failure Modes and Effects Criticality Analysis
FTA	Fault Tree Analysis
HA	Hazard Analysis
ICBM	Intercontinental Ballistic Missile
MDA	Missile Defense Agency
MFD	Multi-Function Display
MIL-STD	Military Standard
MIT	Massachusetts Institute of Technology
OSH	Operating and Support Hazard Analysis
PIC	Pilot in Command
PHA	Preliminary Hazard Analysis
PHL	Preliminary Hazard List
PVI	Pilot Vehicle Interface
RTCA	Radio Technical Commission for Aeronautics
SAR	Safety Assessment Report
SHA	System Hazard Analysis
SSHA	Subsystem Hazard Analysis
STAMP	Systems-Theoretic Accident Model and Processes

STPA	Systems-Theoretic Process Analysis
UAS	Unmanned Aircraft Systems
UCA	Unsafe Control Action
WCAAS	Warning, Caution, Advisory and Alerting System

Bibliography

[1] W. I. Steiglitz, "Engineering For Safety," *Aeronautical Engineering Review,* vol. 7, no. 2, pp. 18-23, 1948.

[2] C. A. Ericson, Hazard Analysis Techniques for System Safety, Wiley, 2005.

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

[4] Department of Defense, "MIL-STD-882E System Safety," Department of Defense, Washington D.C., 2012.

[5] C. Flemming, "Safety-Driven Early Concept Analysis and Development," Ph.D. Dissertation, Dept. of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, 2015.

[6] Plane Crash Info, "Causes of Fatal Accidents by Decade," 31 12 2015. [Online]. Available: http://www.planecrashinfo.com/cause.htm. [Accessed 23 1 2017].

[7] J. W. Vincoli, Basic Guide to System Safety, 3rd Edition, Wiley, 2014.

[8] N. G. Leveson, Engineering a Safer World, Cambridge: The MIT Press, 2011.

[9] W. R. Ashby, An Introduction to Cybernetics, London : Chapman and Hall, 1956.

[10] P. Checkland, Systems Thinking, Systems Practice, New York: John Wiley & Sons, 1981.

[11] N. Weiner, Cybernetics: or the Control and Communication in the Animal and the Machine, Cambridge: MIT Press, 1948.

[12] L. V. Bertalanffy, "An Outline of General Systems Theory," *British Journal of Philosophy of Science,* vol. 1, pp. 134-165, 1950.

[13] N. Leveson, Safeware: Systems Safety and Computers, Waltham, MA: Addison-Wesley, 1995.

[14] C. O. Miller, "A Comparison of Military and Civil Approached to Aviation Systems Safety," *Hazard Prevention,* no. May-June, 1985.

[15] G. Weinberg, An Introduction to General Systems Thinking, New York: John Wiley & Sons, 1975.

[16] S. Ramo, "The Systems Approach," in *Systems Concepts: Lectures on Contemporary Approaches to Systems*, New York, John Wiley & Sons, 1973, pp. 13-32.

[17] S. Pereira, G. Lee and J. Howard, "A System-Theoretic Hazard Analysis Methodology for a Non-Advocate Safety Assesssment of Ballistic Missile Defense System," *Proceeding of the 2006 AIAA Missile Sciences Conference*, no. November, 2006.

[18] SAE ARP 4761, "Guidlines and Methods for Conducting the Safety Assessment Process on Civil Airborne Systems and Equipment," 1996.

[19] A. Hein, D. Horney, M. Mingler and N. Peper, *STPA Analysis of the AH-64 Apache Weapons Systems,* Unpublished MIT Class Project, 2016.

[20] J. Thomas, "Extending and Automating A Systems-Theoretic Hazard Analysis for Requirements Generation and Analysis," Massachusetts Institute of Technology, Cambridge, 2013.

[21] N. Leveson, "Intent Specifications: An Approach to Builiding Human-Centered Specifications," *IEEE Trans. On Software Engineering,* 2000.

[22] D. Montes, "Using STPA to Inform Developmental Product Testing," Ph.D. Dissertaition, Dept. of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, 2016.

[23] N. Leveson, "A Systems Appraoch to Risk Management Through Leading Safety Indicators," *Journal of Reliability Engineering and Safety,* 2017.

[24] A. Ball, "Identification of Leading Indicators for Producibility Risk In Early-Stage Aerospace Product Development," Master's Thesis, Massachussets Institute of Technology, Cambridge, 2015.

[25] J. Rasmussen, "Risk Management in a Dynamic Society: A Modeling Problem," *Safety Science,* vol. 27, no. (2/3), pp. 183-213, 1997.

[26] Department of the Army, "Pamphlet 385-16 System Safety Management Guide," Department of the Army, Washington, 2013.

[27] R. R. Frola and C. O. Miller, "System Safety In Aircraft Acquisition," Logistics Management Institute, Washington, D. C. , 1984.

[28] A. Strafaci, "What does BIM mean for civil engineers?," *CE News,* no. October 2008, pp. 62-65, 2008.

[29] L. A. T. C. Jr., "What's Wrong WIth Risk Matrices?," *Risk Analysis,* vol. 28, no. 2, pp. 497-512, 2008.

[30] M. Stringfellow, "Accident Analysis and Hazard Analysis for Human and Organizational Factors," Ph. D. Dissertation, Massachusetts Institute of Technology, 2010.

[31] K. Chung, "Systems-Theoretic Process Analysis of the Air Force Test Center Safety Management System," Master's Thesis, Massachusetts Institute of Technology, 2014.

Appendix A. Control Diagrams with Labels

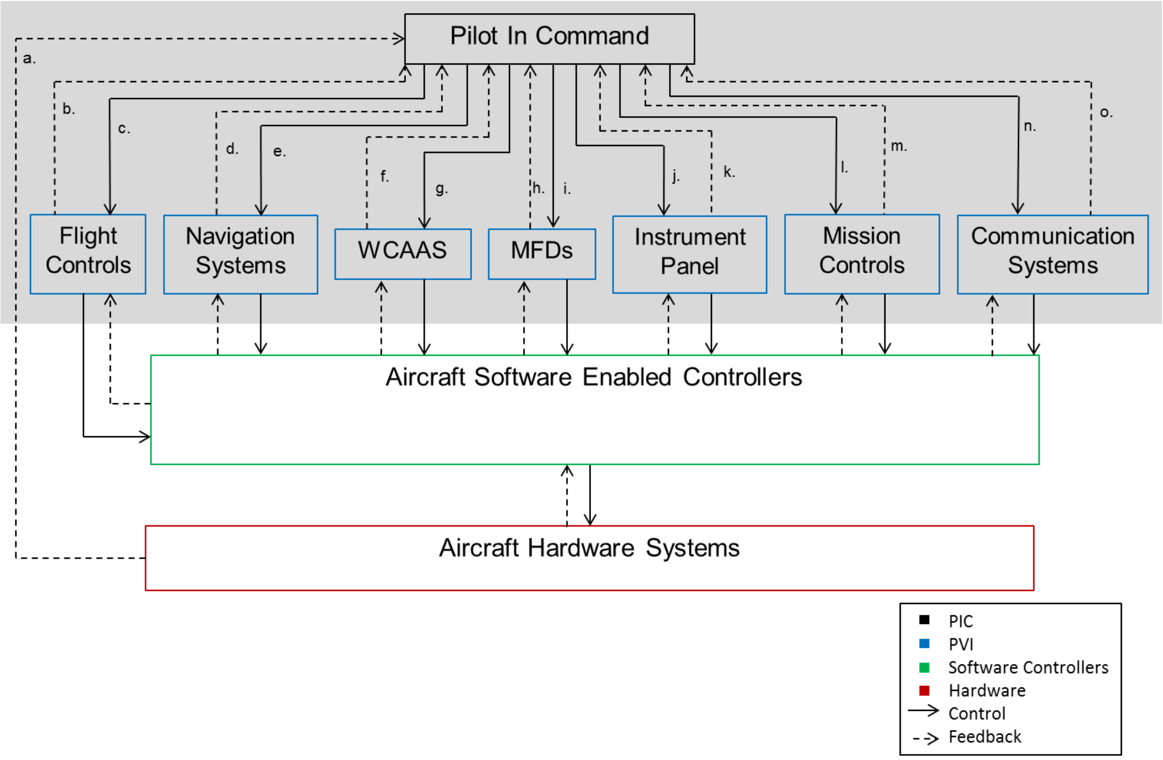


Figure 28. Details of the interactions of the PIC with the different parts of the Pilot Vehicle Interface (PVI) (hardware and software). To make the model more readable, only the details of the PIC and PVI are shown, with the other boxes providing overall context. The blue boxes represent physical interfaces as well as embedded software.

a. **Feedback: Aircraft Hardware Systems → PIC**

Noise from the airframe

Engine noise

Vibrations

Visible battle damage

Hardware status

Odors from aircraft systems

Environmental and system conditions (such as temperature and pressure of the cabin, temperature of the avionics bay, fire and smoke indicators)

b. **Feedback: Flight Controls → PIC**

Haptic flight control feedback

Flight control position

c. **Control Actions: PIC → Flight Controls**

Set aircraft attitude

Set aircraft power

Set aircraft altitude

Set formation shape

Choose emergency response for tethered vehicles

d. **Feedback: Navigation Systems → PIC**

Heading

Waypoints

Aircraft position

Position of known obstacles

Aircraft velocity

Operational Status (i.e. good signal)

e. **Control Actions: PIC → Navigation Systems**

Provide desired route to navigation systems

f. **Feedback: Warning, Caution, Advisory and Alerting System (WCAAS) → PIC**

Notification of equipment malfunction

Alert when flight limits are exceeded

Alert when separation is violated

Notification of enemy fire

g. **Control Actions: PIC → WCAAS**

Acknowledge WCAAS

Set default warning parameters

h. **Feedback: MFDs → PIC**

Heading

Attitude

Engine information (such as RPM, engine temperatures, pressures, etc.)

Altitude

Airspeed

Vertical velocity

Load factor

Mission feeds from other mission actors

Computer performance

Computer software status

Airframe health

Electrical system information

Equipment settings

Temperature data

Radar output

Flight limits

Weather conditions

Position of traffic near aircraft

i. Control Actions: PIC → MFDs

Select feedback to see

Interact with mission feeds

Change equipment settings

Reboot/reload equipment

j. Control Actions: PIC → Instrument Panel

Change instrument settings

k. Feedback: Instrument Panel → PIC

Velocity

Altitude

Vertical Velocity

Heading

Attitude

Engine RPM

Main Rotor Speed

Engine temperatures

Maximum performance capabilities

l. Control Actions: PIC → Mission Controls

Select/deploy countermeasures

Employ defensive avionics (RWR, etc.)

Change aircraft subsystem settings

Reboot/reload aircraft subsystem

m. Feedback: Mission Controls → PIC

Countermeasures available

Aircraft subsystem settings

n. Control Actions: PIC → Communications Systems

Provide communications

o. Feedback: Communications Systems → PIC

Communications from other actors

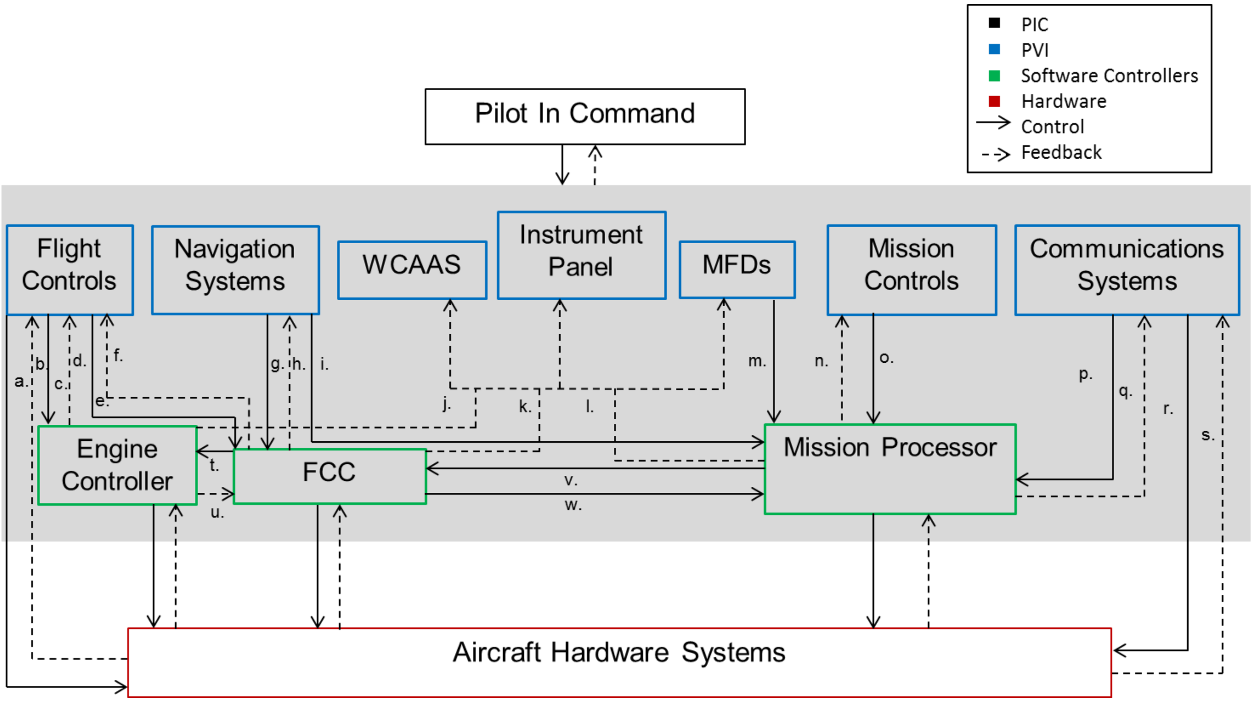


Figure 29. Detailed control structure for Pilot Vehicle Interface (hardware and software) to Aircraft Software-Enabled Controllers (shown in the grey area).

a. Control Actions: Flight Controls → Aircraft Hardware Systems

Actuate directly connected flight systems

b. Feedback: Aircraft Hardware Systems → Flight Controls

State of directly controlled hardware

c. **Control Actions: Flight Controls→ Engine Controller**

Translate engine control inputs to be implemented by the engine controller

d. **Feedback: Engine Controller → Flight Controls**

Engine RPM

Engine temperatures

Maximum performance capabilities

Appropriate throttle state

e. **Control Actions: Flight Controls → FCC**

Translate maneuvering inputs to be implemented by the FCC

f. **Feedback: FCC → Flight Controls**

Aircraft pitch

Aircraft roll

Aircraft yaw

Appropriate flight controls position

Aircraft limits

g. **Control Actions: Navigation Systems → FCC**

Translate navigation inputs from the pilot to be implemented by the FCC

h. **Feedback: FCC → Navigation Systems**

Aircraft velocity

Aircraft position

i. **Control Actions: Navigation Systems → Mission Processor**

Translate navigation inputs to be implemented by the mission processor

j. **Feedback: Engine Controller → WCAAS, MFDs, Instrument Panel**

Engine RPM

Engine temperatures

Maximum performance capabilities

Engine limits

Fuel use

Fuel flow

Computer performance

k. Feedback: FCC → WCAAS, MFDs, Instrument Panel

Aircraft velocity

Main Rotor Speed

Vertical velocity

Altitude

Attitude

Heading

Aircraft flight limits

Computer performance

l. Feedback: Mission Processor → WCAAS, MFDs, Instrument Panel

Potential threats

Communication outputs

Countermeasure state

Subsystem performance

Malfunctioning system alerts

Power use

Power available

Computer performance

m. Control Actions: Instrument Panel → Mission Processor

Translate inputs to be implemented by the mission processor

n. Feedback: Mission Processor → Mission Controls

Countermeasures available

Mission system performance

Mission system state

o. Control Actions: Mission Controls → Mission Processor

Translate mission system inputs to be implemented by the mission processor

p. Control Actions: Communications Systems → Mission Processor

Translate communication inputs from the pilot to be implemented by the mission processor

q. Feedback: Mission Processor → Communications Systems

Incoming messages

Encryption information

Available communication paths

r. **Control Actions: Communications Systems → Aircraft Hardware Systems**

Transmit communications directly through hardware

s. **Feedback: Aircraft Hardware Systems → Communications Systems**

Direct communication outputs

t. **Control Actions: FCC → Engine Controller**

Desired power

Atmospheric Information

u. **Feedback: Engine Controller → FCC, Mission Processor**

Engine RPM

Engine temperatures

Maximum performance capabilities

Power output

Power available

v. **Feedback: Mission Processor → FCC, Engine Controller**

Weight and balance information

w. **Feedback: FCC → Mission Processor**

Flight actuator positions

Aircraft position

Aircraft velocity

Atmospheric information

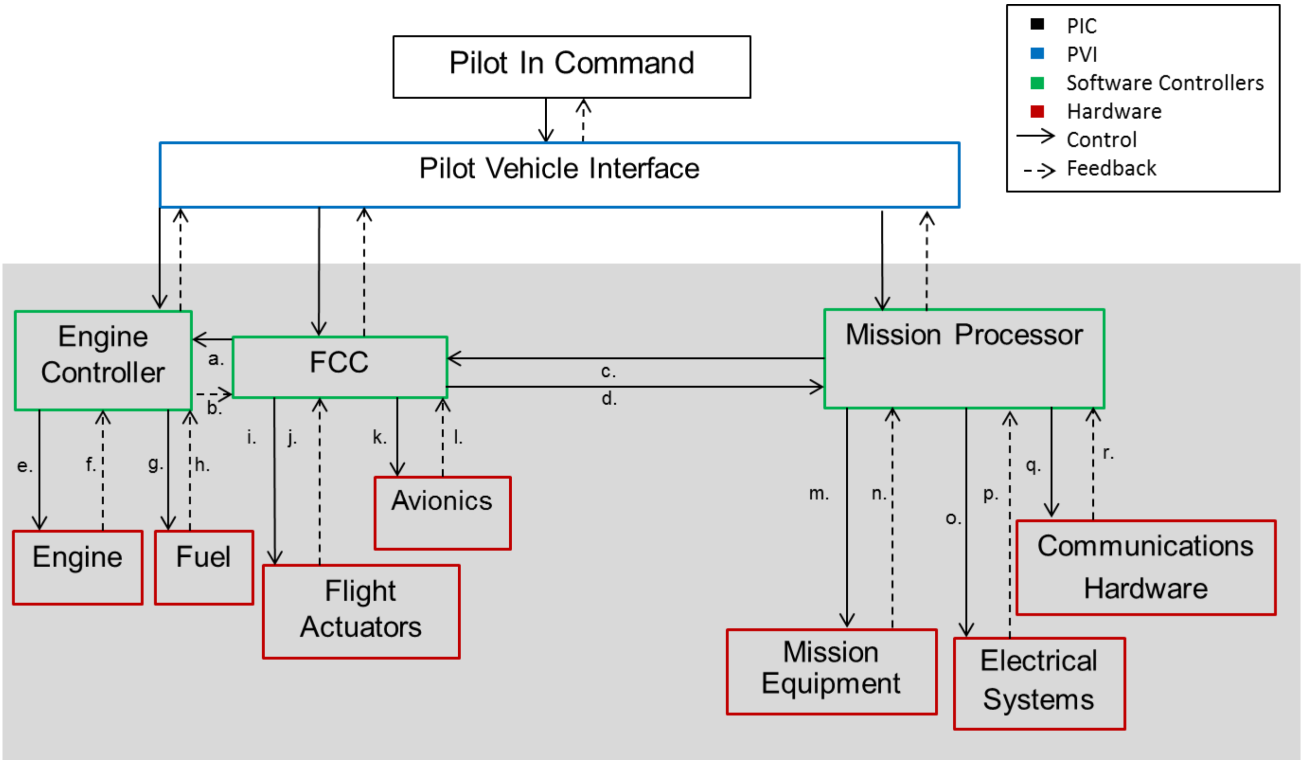


Figure 30. Detailed control structure for Aircraft Software-Enabled Controllers to Aircraft Hardware systems.

a. **Control Actions: FCC → Engine Controller**

Desired power

Atmospheric Information

b. **Feedback: Engine Controller → FCC, Mission Processor**

Engine RPM

Engine temperatures

Maximum performance capabilities

Power output

Power available

c. **Feedback: Mission Processor → FCC, Engine Controller**

Weight and balance information

d. **Feedback: FCC → Mission Processor**

Flight actuator positions

Aircraft position

Aircraft velocity

Atmospheric information

e. **Control Actions: Engine Controller → Engine**

Provide throttle setting

Control bleed air

Start engine command

f. **Feedback: Engine → Engine Controller**

Engine RPM

Engine temperatures

Maximum performance capabilities

Hardware position and state

Atmospheric data

Bleed air usage

g. **Control Actions: Engine Controller → Fuel Systems**

Command fuel flow

h. **Feedback: Fuel Systems → Engine Controller**

Fuel level

Fuel flow

i. **Control Actions: FCC → Flight Actuators**

Provide actuator positions

Provide actuation speed

j. **Feedback: Flight Actuators → FCC**

Hardware position and state

Hydraulic pressure

k. **Control Actions: FCC → Flight sensors**

Provide calibration data

l. **Feedback: Flight sensors → FCC**

Atmospheric data

Aircraft velocity

Aircraft position

Aircraft attitude

Altitude

m. Control Actions: Mission Processor → Mission Equipment

Provide actuator positions

Control power state

Command countermeasures

Control survivability equipment

Control mission function

n. Feedback: Mission Processor → Mission Equipment

Actuation state

Power use

Countermeasures available

Mission function status

o. Control Actions: Mission Processor → Electrical Systems

Control power distribution

p. Feedback: Electrical Systems → Mission Processor

Electrical use

q. Control Actions: Mission Processor → Communications Hardware

Provide outgoing messages

Provide communication settings

Provide desired recipient

Provide encryption information

r. Feedback: Communications Hardware → Mission Processor

Incoming messages

Encryption information

Communication source

Communication type

Power use

Appendix B: Detailed STPA Examples

Controller: Pilot in Command

Controlled Process: Flight Controls

Control Action	Not Providing Causes Hazard	Providing Causes Hazard	Incorrect Timing/ Incorrect Order	Stopped Too Soon/ Applied Too Long
1.Set Aircraft Attitude	UCA 1.1: The PIC does not set aircraft attitude when the trimmed aircraft state is causing the aircraft to deviate from its flight plan. (H1)	UCA 1.2: The PIC sets an incorrect aircraft attitude causing the aircraft to violate separation minimums. (H2) UCA 1.3: The PIC sets an aircraft attitude that is not achievable. (H2)	UCA 1.4: The PIC changes aircraft attitude at a rate that will damage the airframe. (H3)	UCA 1.5: The PIC changes the aircraft attitude at too high or too small a magnitude when there is an obstacle nearby. (H1) UCA 1.6: The PIC changes aircraft attitude too much or too little when the aircraft is close to its flight limits. (H2)
2. Set Aircraft Power	UCA 2.1: The PIC does not adjust aircraft power when there is an obstacle approaching the aircraft's position. (H1)	UCA 2.2: The PIC sets the aircraft power too high causing the aircraft to exceed VNE. (H3) UCA 2.3: The PIC sets the aircraft power to a value that does not allow it to complete the	UCA 2.4: The PIC changes aircraft power at a rate that will damage the airframe. (H3)	N/A

		selected maneuver. (H2)		
3. Set Aircraft Altitude	N/A	UCA 3.1: The PIC sets the aircraft altitude that exceeds aircraft performance capabilities. (H3) UCA 3.2: The PIC sets an aircraft altitude that will violate separations with an obstacle. (H1)	N/A	N/A
4. Set Formation Shape	UCA 4.1: The lead aircraft PIC does not set a new formation shape when needed. (H1, 2)	UCA 4.2: The lead aircraft PIC sets an unsafe formation shape for the current environment. (H1, 2)	N/A	N/A
5. Choose Emergency Reaction for Tethered Vehicles	UCA 5.1: The PIC does not choose an emergency reaction for the tethered vehicles when an emergency state arises. (H1, 2, 3)	UCA 5.2: The PIC chooses an emergency reaction for the tethered vehicles that is not appropriate for the emergency state. (H1, 2, 3)	UCA 5.3: The PIC chooses an emergency reaction for the tethered vehicles too long after the emergency occurs for it to be effective. (H1, 2, 3)	UCA 5.4: The PIC stops the emergency reaction before the emergency has been resolved. (H1, 2, 3)

UCA 1.1: *The PIC does not set aircraft attitude when the trimmed aircraft state is causing the aircraft to deviate from its flight plan.* (H1)

Example Causal Scenarios for UCA 1.1a: The PIC does not set aircraft attitude when the trimmed aircraft state is causing the aircraft to deviate from its flight path because the PIC believes that the aircraft is on the desired flight path, i.e., his/her process model is flawed. This could occur if:

1. The PIC does not receive detailed enough feedback through the PVI to detect small deviations from the flight plan. These small deviations become greater over time and the

aircraft could violate separation with an obstacle. This problem can be compounded in tethered scenarios where the PIC is responsible for multiple vehicles and cannot easily check clearances for the tethered vehicles.

2. The PIC did not program a flight plan into the aircraft systems and is unable to project a flight path for the aircraft in the airspace because of degraded conditions or a high workload. This results in disorientation and a loss of aircraft control or violation of separation standards.
3. The aircraft is incorrectly indicating that it is following the desired flight path. This could occur if the navigation system or PVI supplies inaccurate or insufficient feedback to the PIC due to degraded operation, enemy kinetic/cyber attack, or poor design.
4. The PIC believes that autopilot is engaged and the aircraft computer systems are controlling pitch. This mode confusion could occur if there is not clear feedback to the PIC about the autopilot's state and/or the autopilot does not alert the pilot if it changes state.
5. Unauthorized communication from a second subsystem provides an incorrect state to the PIC. This could occur if subsystems are allowed to send arbitrary messages to displays leveraged by the PIC to inform his or her process model.
6. Malformed communication from sensors leads to incorrectly indicating the aircraft is following the desired path.
7. Adversaries purposely provide incorrect input to the pilot by altering the feedback to the displays.

Example requirements:

a. Navigation systems and interfaces shall allow for navigation with error less than TBD miles in manual flight modes.
b. The navigation systems and PVI shall be monitored for faults at TBD Hz to ensure that they are updating.
c. Navigation accuracy shall be confirmed through multiple independent sensors.
d. The PIC shall be alerted when the aircraft deviates from the flight plan by TBD feet.
e. The PIC shall be alerted if there is an object within TBD feet of separation minimums.
f. The aircraft shall provide the PIC with clear feedback to indicate what responsibilities the computer systems are currently taking.
g. The aircraft shall alert the pilot if there are uncommanded mode changes by the computer systems.
h. The aircraft shall monitor the PIC's state of arousal and simplify the feedback presented if the PIC has degraded abilities and is unable to operate under a high workload.
i. The PIC shall be provided with separation information about tethered vehicles and a visualization of the entire formation relative to the flight environment.
j. A minimal flight plan shall be provided for all missions so that the aircraft can be reoriented to the flight path if the PIC becomes disoriented.
k. Data sources for informational displays shall be specified. Integrity validation shall be performed on all data sources.
l. Explicit specification of data formats shall be provided for informational displays so that the displays can be verified to correctly parse data and correctly respond to malformed data.

Causal Scenario 1.1b: The PIC realizes that the aircraft is deviating from its flight plan but wants to make infrequent larger corrections rather than continually correcting the flight path because the pilot is overloaded. These larger corrections result in an unsafe aircraft state such as violation of separation minimums.

Example Requirements:

a. The aircraft shall have autopilot systems that allow for precise following of the flight plan in order to free up the pilot for other mission related duties.
b. The aircraft systems shall continuously project a flight path for the aircraft and check that path for separation violations.

UCA 1.2: The PIC sets an incorrect aircraft attitude causing the aircraft to violate separation minimums. (H2)

Causal Scenario 1.2a: The PIC sets an incorrect aircraft attitude but believes he set the correct attitude. This could occur if:

1. The aircraft is operating and a rapidly changing environment and the PIC must take in information about the surroundings and respond with appropriate controls. He is unable to predict the correct control input for the situation and inputs an unsafe attitude command. An example of such an environment is landing on a pitching boat deck on rough seas.
2. The PIC receives incorrect or insufficient feedback from aircraft state sensors that cause the PIC to adjust the aircraft based on inaccurate information.
3. There has been battle damage to the aircraft that changes the aircraft handling characteristics and response to input.
4. The PIC receives incorrect or insufficient feedback from the controls that causes the PIC to respond incorrectly.
5. The aircraft is operating in a degraded environment and the feedback that the PIC typically relies on is not available.

Example Requirements:

a. The feedback and control loop shall have high enough gain to allow the PIC to accurately control the aircraft in dynamic conditions. (Margins TBD)
b. The FCC shall assist the PIC in controlling the aircraft in scenarios that require reaction times of less than TBD seconds.
c. The aircraft shall provide multiple sources of feedback that are checked for accuracy against one another.
d. Aircraft health monitoring systems shall notify the PIC of any battle damage that occurs on the aircraft and how it might affect aircraft performance.

e. The flight control design group shall perform user testing with pilots to ensure that the flight controls provide sufficient and useful feedback.
f. The aircraft shall provide the PIC with backup sources of feedback.
g. Feedback that is available in degraded conditions shall be presented in the same way that the primary feedback is presented so the pilots can easily use the feedback.

Causal Scenario 1b1.2: The PIC sets an incorrect aircraft attitude the causes the aircraft to violate separation with terrain because the aircraft is operating in an emergency state and the PIC must execute an emergency landing in order to save the aircraft. The PIC chooses control actions that will minimize harm to the aircraft and its occupants.

Example Requirements:

a. The PIC shall be able to actuate the control surfaces even if the primary control path is severed.

UCA 1.4: The PIC changes aircraft attitude at a rate that will damage the airframe. (H3)

Causal Scenario 1.4a: The PIC changes aircraft attitude at a rate that will damage the airframe but believes the attitude is changing at a safe rate. This could occur if:

1. The PIC is at a high level of arousal and uses greater than normal force to control the aircraft. This results in the pilot putting in more aggressive control inputs and damaging the airframe.
2. There is not sufficient feedback to the PIC to inform him of the aircraft dynamics that he is commanding. The lack of feedback causes over-control of the aircraft.
3. The aircraft is operating in degraded conditions and the normal feedback that the PIC relies on to determine appropriate inputs is not available.

Example Requirements:

a. The aircraft shall monitor the PIC's state of arousal and adjust the feedback force of the controls if the PIC quickly changes to a state of high arousal.
b. Control feedback shall vary with control position and rate to alert the PIC of the aircraft limits.
c. The aircraft shall supply additional feedback to augment the PIC's mental model in degraded conditions.

Causal Scenario 1.4b: The PIC changes the aircraft attitude at a rate that will damage the airframe in order to avoid violating separations or losing control of the aircraft.

Example Requirements:

a. The aircraft shall provide the PIC with details concerning the damage so that the PIC can avoid maneuvers that would aggravate any existing damage to the airframe if the airframe becomes damaged.

b. The WCAAs shall provide advanced warning when the aircraft is likely to violate separations or depart controlled flight so that the PIC can recover without damaging the aircraft.

Causal Scenario 1B: The PIC sets appropriate aircraft pitch but the command is not correctly implemented. This could occur if there is a hardware failure or software design flaw in the control path between the PIC and the aircraft hardware systems. This could consist of:

1. A failure of the flight controls or their connection to the software based PVI.
2. An error in communication between the PVI and FCC.
3. An error in communication between the FCC and the flight actuators.
4. A mechanical failure of the flight actuators.
5. A delay in the control path that results in an unsafe maneuver.

<u>UCA 3.1: The PIC sets the aircraft altitude that exceeds aircraft performance capabilities. (H3)</u>

Causal Scenario 3.1a: The PIC sets an aircraft altitude that exceeds aircraft performance capabilities because he believes the altitude is achievable within the performance envelope. This could occur if:

1. The ambient conditions have changed from what was planned for when the mission began and resulted in a narrower flight envelope. The PIC did not get updated environmental and performance conditions and was unable to adjust his mental model of the aircraft to match the operating conditions. This situation is unsafe if the aircraft is trying to navigate terrain with an altitude that exceeds the aircraft's performance in the current atmospheric conditions or if the aircraft is attempting to climb to avoid an airborne obstacle.
2. The PIC is controlling a flight of tethered aircraft and one or more of the tethered aircraft has lower performance capabilities than the lead aircraft due to a heavier load or degraded performance. The PIC sets the altitude for the flight based on the capabilities of their own aircraft and the tethered vehicle is not able to achieve the set altitude.
3. The PVI allows for altitude inputs outside of the operational envelope of the aircraft.
4. Ambient conditions prevent the PIC from perceiving current altitude. Feedback from altitude indicators incorrectly indicate that the aircraft is above or below desired operational altitude, causing the PIC to adjust the altitude that exceeds performance envelope. This may be caused by faulty hardware, malicious logic, or design flaws; the PIC is wearing night-vision goggles impairing ability to perceive altitude correctly; or the pilot is in brown-out conditions.

<u>Example Requirements</u>:

a. The PIC shall be provided with real-time performance capabilities that incorporate atmospheric data and present updated flight limits.
b. The WCAAs shall alert the PIC if the aircraft is approaching its maximum performance capabilities.

c. The aircraft shall continuously check the flight path against the ambient conditions to determine if route changes are necessary to ensure a safe route.
d. The PIC shall have independent indicators of current operational status and operational flight envelope to overcome potential compromised sensors.

Causal Scenario 3.1b: The PIC sets an aircraft altitude that exceeds aircraft performance capabilities because the aircraft is in an emergency state and the only way to save the aircraft is to exceed the performance margins. This could occur if:

1. The system is in an actual emergency state.
2. The PIC incorrectly perceives system to be in an emergency state caused by malformed feedback from subsystems.

Example Requirements:

a. The aircraft shall provide multiple feedback modalities to alert the PIC when they are approaching the performance limits of the aircraft.
b. The aircraft shall allow the PIC to exceed the performance limits of the aircraft in emergency situations in order to save the aircraft and its occupants but make sure that the pilot is aware that the performance limits are being exceeded [without diverting needed pilot sensory resources)
c. The aircraft shall provide multiple checks to ensure that expression of emergency state by aircraft can be verified.

Causal Scenario 3B: The PIC sets a safe altitude but it is not correctly implemented. This could occur if there is a hardware failure or software design flaw in the control path between the PIC and the aircraft hardware systems. This could consist of:

1. A failure of the flight controls or their connection to the software based PVI.
2. An error in communication between the PVI and FCC.
3. A software error in the FCC.
4. An error in communication between the FCC and the flight actuators.
5. A mechanical failure of the flight actuators.
6. There is a delay in the control path that results in an unsafe maneuver.

Example Requirements:

a. There shall be independent redundant control paths for safety critical functions.
b. The aircraft shall have backup flight control systems that will allow for controlled flight if the primary actuators fail.
c. The aircraft shall check system functionality and alert the PIC if systems are operating in a degraded state.
d. In multi-station cockpits, each control station shall be connected using a separate control path to create control redundancy.
e. An independent system shall check for delays in operation and alert the PIC if delays are detected.

UCA 4.1: The lead aircraft PIC does not set a new formation shape when needed. (H1, 2)

Causal Scenario 4.1a: The lead aircraft PIC does not set a new formation shape when needed because the PIC believes the current formation shape is sufficient. This could occur if:

5. The lead aircraft PIC is not able to predict future states of the formation and therefore does not know that a new formation shape is needed to avoid a conflict or unsafe flight configuration.
6. The lead aircraft PIC is task saturated and cannot generate an accurate process model of the entire tethered formation and the environment they are operating in.
7. There is not sufficient feedback from the tethered aircraft for the lead aircraft PIC to determine the best formation shape for the situation at hand.
8. There is malformed feedback from the tethered aircraft for the lead aircraft PIC to determine the best formation shape for the situation at hand. This may be in the form of incorrect position information, dropped feedback, communication with tethered aircraft has been lost, or malformed data that is not displayable by system.

<u>Example Requirements</u>:

f. The lead aircraft PIC shall be provided with feedback to predict future states of the formation. It is a difficult cognitive task to predict the future state of multiple vehicles so it is likely that predictive aids will be required.
g. Studies shall be performed to determine how pilots will respond while flying a formation with tethered aircraft. The system shall be designed to keep the workload within the PIC’s capabilities even during emergency situations.
h. There shall be enough feedback from the tethered aircraft to the lead PIC to allow the lead PIC to make informed decisions about the formation.
i. System shall indicate to PIC current communication status between lead aircraft and tethered aircraft.
j. The system shall indicate the last known good information, and corresponding age of information to the PIC in the lead aircraft.

<u>UCA 4.2: The lead aircraft PIC sets an unsafe formation shape for the current environment. (H1, 2]</u>

Causal Scenario 4.2a: The lead aircraft PIC sets an unsafe formation shape for the current environment because they believe the new formation shape is the best shape for the environment. This process model flaw could arise if:

5. The lead aircraft does not have an accurate model of the current environment because there are degraded conditions or aircraft sensors.
6. The lead aircraft is not able to predict future states of the formation and therefore does not know that the prescribed formation shape will be unsafe.
7. The lead aircraft receives inaccurate/partial/malformed feedback about the current environment from their aircraft, the tethered aircraft, or outside parties.

<u>Example Requirements</u>:

e. Formation shapes and control modes must be designed for degraded conditions that could lead to a hazard.
f. The lead aircraft shall be provided with sufficient feedback to predict future states of the formation. It is a difficult cognitive task to predict the future state of multiple vehicles so it is likely that predictive aids will be required.
g. The lead aircraft shall receive an alert if the aircraft sensors are not providing accurate feedback about the environment. In a tethered situation, feedback from all aircraft in the formation shall be integrated and compared to increase situational awareness and allow for better error checking.

Causal Scenario 4.2b: The lead aircraft provides an unsafe formation shape for the current environment because the formation is in a dynamic environment and the lead aircraft is not able to provide adequate control for the formation.

Example Requirements:

d. Tethered configurations shall avoid rapidly changing environment.
e. Tethered aircraft shall have independent threat detection and terrain avoidance systems that allow them to react to environmental hazards without lead aircraft input.
f. Maintaining separation with other aircraft in the formation shall take priority over avoiding other hazards in the environment. (If tethered aircraft are ever certified to transport humans, the hazard avoidance logic shall be updated to minimize the likelihood of harm to the occupants and other aircraft in the formation.)

Causal Scenario 4B: The PIC sets a safe formation shape for the tethered formation but it is not correctly implemented or followed. This could occur if:

8. There is a failure of the flight controls or their connection to the software based PVI.
9. There is a miscommunication between the software based PVI and the mission computer which is responsible for communicating with the tethered vehicles.
10. There is a hardware failure in the communication link between the lead aircraft and the tethered vehicles.
11. There is a malfunction in one or more of the tethered vehicles that does not allow them to reach the desired formation shape.
12. There is a delay in the control path causing the new formation shape to be implemented too late.
13. Malfunction in the communications between the PIC and tethered aircraft.
14. Compromised control path drops, interferes with, or manipulates the PIC commands to the mission system and/or tethered aircraft, despite receiving feedback that a new formation was commanded.

Example Requirements:

i. The WCAAS shall alert the PIC if one of the tethered vehicles is operating in a degraded condition.
j. There shall be independent backup communication systems that can be used to maintain communication within the formation in case the primary communication channels are lost.
k. Each aircraft shall have an independent loss of link plan that corresponds to its position in the formation and allows it to exit the formation safely.

l. There shall be adequate sensors on each aircraft to allow them to safely navigate the airspace to a safe landing position without control by the lead aircraft.
m. Each tethered aircraft shall have a loss of link plan that is updated throughout the mission, which allows for the aircraft to safely land as soon as possible.
n. All aircraft shall be able to autonomously coordinate with other air traffic to avoid conflict.
o. All aircraft shall be able to find a suitable landing spot in unfamiliar areas.
p. The tethered vehicles shall send a message to the lead aircraft when they receive and act on commands. If a confirmation message isn't received within TBD seconds, the PIC must be alerted.

UCA 5.1: The PIC does not choose an emergency reaction for the tethered vehicles when an emergency state arises. (H1, 2, 3)

Causal Scenario 5.1a: The PIC does not choose an emergency reaction for the tethered vehicles when an emergency state arises because he believes that the aircraft are operating as intended and has not realized that an emergency state exists. This could occur if:

1. One or more of the aircraft is operating in a degraded state but the PIC was not alerted.
2. There is a breakdown in communication between the aircraft and an alert that a tethered vehicle is operating in a degraded state does not reach the lead aircraft.
3. The PIC is operating under an excessive workload and does not notice warnings that the formation is in an emergency state.

Example Requirements:

a. There shall be fault testing in the aircraft systems that trigger alerts when aircraft components are not operating correctly.
b. The tethered aircraft shall provide status updates throughout the mission as TBD Hz.
c. If the lead aircraft does not receive a status update from a tethered aircraft, the PIC shall be alerted.
d. Studies shall be performed to determine how pilots will respond while flying a formation with tethered aircraft.
e. The system shall be designed to keep the workload within the PIC's capabilities even during emergency situations.

Causal Scenario 5.1b: The PIC does not choose an emergency reaction for the tethered vehicles when an emergency state arises because the lead vehicle has been compromised and the PIC is no longer in control of the formation.

Example Requirements:

1. If the lead aircraft is lost, the formation shall be programmed to enter a safe state.
2. Each aircraft shall have an independent loss of link plan that corresponds to its position in the formation that allows it to exit the formation safely.

3. There shall be adequate sensors on each aircraft to allow them to safely navigate the airspace to landing without control by the lead aircraft.
4. Each tethered aircraft shall have a loss of link plan that is updated throughout the mission, which allows for the aircraft to safely land as soon as possible.
5. All aircraft shall be able to autonomously coordinate with other air traffic to avoid conflict.
6. All aircraft shall be able to find a suitable landing spot in unfamiliar areas.

UCA 5.2: The PIC chooses an emergency reaction for the tethered vehicles that is not appropriate for the emergency state. (H1, 2, 3)

Causal Scenario 5.2a: The PIC chooses an emergency reaction for the tethered vehicles that is not appropriate for the emergency state because he does not have an accurate understanding of the emergency state and/or the emergency reactions. This could occur if:

1. There is a breakdown in communication between the aircraft and the PIC does not get sufficient information from the other aircraft in the formation to understand the issue at hand.
2. The PIC is operating under an excessive workload and is unable to create an accurate process model of the formation.
3. The PIC does not have sufficient information about the current environmental conditions to choose the best emergency reaction because the flight is operating in degraded conditions and/or the environmental sensors are not providing accurate data.
4. There are too many emergency reactions for the PIC to pick the best reaction for each emergency situation.

Example Requirements:

a. Checks shall be enforced to ensure that messages are accurately passed between aircraft systems.
b. There shall be independent backup communication systems that can be used to maintain communication within the formation in case the primary communication channels are lost.
c. Studies shall be performed to determine how pilots will respond while flying a formation with tethered aircraft. The system must be designed to keep the workload within the PIC’s capabilities even during emergency situations.
d. The lead aircraft shall receive an alert if the aircraft sensors are not providing accurate feedback about the environment.
e. In a tethered situation, feedback from all aircraft in the formation shall be integrated and compared to increase situational awareness and allow for better error checking.
f. Emergency reactions shall be designed to encompass as many scenarios as possible to avoid excessively complicated procedures.

Causal Scenario 5B: The PIC sets a safe emergency response for the tethered formation but it is not correctly implemented. This could occur if:

1. There is a failure of the flight controls or their connection to the software based PVI.
2. There is a miscommunication between the software based PVI and the mission computer responsible for communicating with the tethered vehicles.
3. There is a hardware failure in the communication link between the lead aircraft and the tethered vehicles.
4. There is a malfunction in one or more of the tethered vehicles that does not allow them to perform the desired emergency reaction.
5. There is a delay in the control path causing the emergency response to be implemented too late.

Example Requirements:

a. The WCAAs shall alert the PIC if one of the tethered vehicles is operating in a degraded condition.
b. There shall be independent backup communication systems that can be used to maintain communication within the formation in case the primary communication channels are lost.
c. Each aircraft shall have an independent loss of link plan that corresponds to its position in the formation that allows it to exit the formation safely.
d. There shall be adequate sensors on each aircraft to allow them to safely navigate the airspace to landing without control by the lead aircraft.
e. Each tethered aircraft shall have a loss of link plan that is updated throughout the mission that allows for the aircraft to safely land as soon as possible.
f. All aircraft shall be able to autonomously coordinate with other air traffic to avoid conflict.
g. All aircraft shall be able to find a suitable landing spot in unfamiliar areas.
h. The tethered vehicles shall send a message to the lead aircraft when they receive and act on commands. If a confirmation message isn’t received within TBD seconds, the PIC must be alerted.

Controller: Pilot in Command

Controlled Process: Navigation Systems

Control Action	Not Providing Causes Hazard	Providing Causes Hazard	Incorrect Timing/ Incorrect Order	Stopped Too Soon/ Applied Too Long
6. Provide Desired Route to Navigation Systems	UCA 6.1: The PIC does not provide their desired route to the navigation system when the aircraft is using autopilot systems. (H1) UCA 6.2: The pilot does not provide their desired route to the navigation systems when they are tethering aircraft. (H1, 2)	UCA 6.3: The PIC provides an unsafe route to the navigation systems. (H1, 2)	N/A	N/A

UCA 6.3: The PIC provides an unsafe route to the navigation systems. (H1, 2)

Causal Scenario 1.3a: The PIC provides an unsafe route to the navigation systems because he does not realize that the route is not safe. This could occur if:

1. The weather degrades from the pre-mission prediction making the route unsafe.
2. There are unanticipated obstacles on the route that interfere with the flight path of the vehicles.
3. Changing conditions make terrain impassable that was navigable in the assumed mission conditions. An example is a high mountain pass that can only be navigated when the temperature is below a certain threshold.
4. Enemy forces have moved into position along the route and present an unexpected threat to the aircraft.

Example Requirements:

a. The aircraft shall be able to receive weather updates from several sources including at least limited on-board sensors.
b. The avionics suite shall incorporate on-board sensors that can detect obstacles in degraded visual conditions.
c. The PIC shall be provided with real-time feedback of ambient conditions.
d. The PIC shall be provided with real-time performance capabilities the incorporate atmospheric data and present updated flight limits.

e. The WCAAS shall alert the PIC if the aircraft is approaching its maximum performance capabilities.
f. The aircraft shall continuously check the flight path against the ambient conditions to determine if route changes are necessary to ensure a safe route.
g. The aircraft shall be able to keep consistent communication contact with intelligence units who have up to date information concerning enemy forces.

Causal Scenario 6B: The PIC sets a safe route but the route is not accepted or implemented by the navigation systems. This could occur if:

1. Enemy forces are jamming GPS rendering the primary navigation systems ineffective.
2. The interface between the PIC and the navigation systems is not functioning properly.
3. There is a miscommunication between the software based PVI and the mission computer responsible for navigation.
4. There is a delay in the navigation system that inhibits safe aircraft control.

Controller: Pilot in Command

Controlled Process: Multifunction Displays

Control Action	**Not Providing Causes Hazard**	**Providing Causes Hazard**	**Incorrect Timing/ Incorrect Order**	**Stopped Too Soon/ Applied Too Long**
7. Select Feedback to See	UCA 7.1: The PIC does not select mission specific feedback to see before the mission begins. (H1, 2)	UCA 7.2: The PIC selects inadequate feedback to see during the mission. (H1, 2, 3)	N/A	N/A

UCA 7.1: The PIC selects inadequate feedback to see during the mission. (H1, 2, 3)

Causal Scenario 1.1a: The PIC selects inadequate feedback to see during the mission because the PIC does not believe they will need feedback other than what they selected. This could occur if:

1. The mission scope changes and additional feedback is required for the mission to be executed.
2. There is an emergency situation that requires additional information that is not easily accessed given the PIC’s choice of feedback to display on the MFD.

Example Requirements:

a. The MFD interface shall allow for feedback to be reconfigured and changed during a mission.

b. There shall be pre-configured displays that help pilots access relevant data in emergency and time critical scenarios.

Causal Scenario 7.1b: The PIC selects unsafe feedback to see during the mission because the MFD cannot support all of the feedback that the PIC wishes to see during the mission simultaneously. This causes the PIC to switch between displays during the mission, which increases workload and decreases attentiveness.

Example Requirements:

1. Feedback shall be scalable and presentable in different forms so that pilots can change the display to suite their mission needs.

Controller: Pilot in Command

Controlled Process: Warning Caution and Advisory System

Control Action	**Not Providing Causes Hazard**	**Providing Causes Hazard**	**Incorrect Timing/ Incorrect Order**	**Stopped Too Soon/ Applied Too Long**
8. Acknowledge WCAA	UCA 8.1: The PIC does not acknowledge a WCAA during a mission. (H1, 2, 3)	UCA 8.2: The PIC acknowledges a WCAA without taking action to address the advisory. (H1, 2, 3)	UCA 8.3: The PIC acknowledges a WCAA too late, after taking action will help. (H1, 2, 3)	N/A
9. Set Default Warning Parameters	UCA 9.1: The PIC does not set default warning parameters before starting a mission. (H1, 2, 3) UCA 9.2: The PIC does not update default warning parameters when transitioning to a new mission environment. (H1, 2, 3)	UCA 9.3: The PIC sets unsafe default warning parameters that do not give the PIC enough time to react to an emergency. (H1, 2, 3)	N/A	N/A

UCA 8.1: The PIC does not acknowledge a WCAA during a mission. (H1, 2, 3)

Causal Scenario 1.1a: The PIC does not acknowledge a WCAAS during a mission because he is not aware that a warning has been issued. This could occur if the warning signal was too weak to be noticed in the mission environment.

Example Requirements:

a. WCAAs shall be expressed via multiple modalities.
b. WCAAs shall increase in intensity if they are not acknowledged.
c. The avionics integration team shall perform human interface testing in simulated combat environments to determine the effectiveness of the WCAAS.

Causal Scenario 8.1b: The PIC does not acknowledge a WCAA during a mission because the workload in the cockpit is too high to have time to acknowledge WCAAS alerts.

Requirement:

a. PICs shall be able to acknowledge and take action on warnings within TBD seconds of them sounding.

UCA 8.3: The PIC acknowledges a WCAA too late, after taking action will help. (H1, 2, 3)

Causal Scenario 1.3a: The PIC acknowledges a WCAA too late because he did not notice or have time to acknowledge the warning any sooner. This could occur if:

1. The WCAA was not noticeable in the mission environment until it was too late to act on.
2. The WCAA was masked by other notifications and was not displayed until it was too late to act on.

Example Requirements:

a. WCAAs shall increase in intensity and employ multiple modalities if they are not noticed and are safety critical. WCAAs shall be organized by importance and criticality. This could vary based on aircraft conditions.

Causal Scenario 8B: The PIC acknowledges a WCAA but the system does not correctly recognize the acknowledgment and the alert continues to sound. This could occur if:

1. There is a failure of the hardware based PVI or its connection to the software based PVI.
2. There is a miscommunication between the software based PVI and the mission computer which is responsible for communicating the WCAAS.
3. There is a delay in the system which causes the PIC to believe that their action was not registered when it actually was. This causes the PIC to spend extra time and effort trying to fix an issue that has already been resolved.

UCA 9.2: The PIC does not update default warning parameters when transitioning to a new mission environment. (H1, 2, 3)

Causal Scenario 9.2a: The PIC does not update default warning parameters when transitioning to a new mission environment because he believes the old parameters are still appropriate. This could occur if:

1. The PIC does not realize that they are transitioning into a new mission environment because they lack mission feedback.
2. The PIC does not realize that the new mission environment requires different alert parameters.

Example Requirements:

a. Minimum alert parameters shall be programmed to ensure there is always an alert parameter programmed.
b. The aircraft shall provide the PIC with multiple sources of mission feedback.
c. Communication systems shall be able to interface with all possible mission actors.

Causal Scenario 9B: The PIC sets appropriate warning parameters but the system does not accept the parameters. This could occur if:

1. There is a failure of the hardware based PVI or its connection to the software based PVI.
2. There is a miscommunication between the software based PVI and the mission computer which is responsible for communicating the WCAAs
3. There is a delay in the updating of the warning parameters and the incorrect parameters are still set during a flight transition.

Controller: Mission Processor

Controlled Process: Communications Hardware

Control Action	Not Providing Causes Hazard	Providing Causes Hazard	Incorrect Timing/ Incorrect Order	Stopped Too Soon/ Applied Too Long
10. Provide Outgoing Messages	UCA 10.1: The mission processor does not provide outgoing messages to be sent by the communications hardware when the outgoing information is needed to avoid a violation of separation (H1)	UCA 10.2: The mission processor provides incorrect or inaccurate outgoing messages to the communications hardware to be sent. (H1)	UCA 10.3: The mission processor provides outgoing messages to the communication hardware in the incorrect order or at the wrong time. (H1)	N/A

UCA 10.1: The mission processor does not provide outgoing messages to be sent by the communications hardware when the outgoing information is needed to avoid a violation of separation (H1)

Causal Scenario 10.1a: The mission processor does not provide outgoing messages to be sent by the communications hardware because it has a flawed process model of the mission environment. This could occur if:

1. The mission processor is not receiving adequate information from the sensors and subsystems to generate the messages that need to be sent in order to maintain separations.
2. The mission processor does not receive a notification that an unsafe state has arisen and it needs to pass a message to other actors in the airspace to maintain safe flight.
3. The mission processor does not receive a command from a higher level controller to generate and send messages to the communication hardware.
4. The mission processor gets incorrect feedback that the communication hardware is not functioning.
5. The mission processor receives an incorrect command from a higher level controller not to send messages to the communication hardware. (This is a result of malicious logic)

Example Requirements:

a. If the aircraft is unable to determine its state because of malfunctioning sensors or subsystems, it shall broadcast this malfunction to surrounding aircraft so that they can heighten their awareness.
b. The mission processor shall be programmed to send aircraft state information to other actors in the airspace if an obstacle comes within TBD distance of the aircraft. (The separation criteria will likely need to be adjustable to allow for formation flight and discreet operation in enemy territory.)
c. Safety critical messages shall be sent automatically in order to maintain separations.
d. The mission processor shall seek out other communication avenues if a hardware system malfunctions. Higher level controllers must be notified of the malfunction.

UCA 10.3: The mission processor provides outgoing messages to the communication hardware in the incorrect order or at the wrong time. (H1)

Causal Scenario 1.3a: The mission processor provides outgoing messages to the communications hardware in the incorrect order because it has a flawed process model of the mission environment. This could occur if:

1. The mission processor is receiving information from the sensors and subsystems in a delayed or mismatched manner which causes the messages to be delayed or incorrect.
2. The mission processor receives a delayed command to send outgoing messages.

Example Requirements:

a. Information exchanged within the aircraft shall be time-stamped in order to allow for temporal checks.

Causal Scenario 10B: The mission processor sends outgoing messages as designed but they are not correctly sent. This could occur if:

1. The signal does not reach the communication hardware or the signal is changed en route.
2. The communication hardware malfunctions.

Controller: FCC
Controlled Process: Engine Controller

Control Action	**Not Providing Causes Hazard**	**Providing Causes Hazard**	**Incorrect Timing/ Incorrect Order**	**Stopped Too Soon/ Applied Too Long**
11. Set desired power.	UCA 11.1: The FCC does not request the desired power level from the engine controller when the engines are turned on. (H1, 2, 3)	UCA 11.2: The FCC requests the incorrect power level from the engine controller. (H1, 2, 3)	UCA 11.3: The FCC requests the desired power setting from the engine controller with a time delay. (H1, 2, 3)	UCA11.4: The FCC stops requesting the desired power setting before the engines shut down. (H1, 2, 3)
12. Provide atmospheric information	UCA 12.1: The FCC does not provide the engine controller with atmospheric information when the engines are turned on. (H1, 2, 3)	UCA 12.2: The FCC provides incorrect atmospheric information to the engine controller. (H1, 2, 3)	UCA 12.3: The FCC provides outdated atmospheric information to the engine controller. (H1, 2, 3)	N/A

UCA 11.1: The FCC does not request the desired power level to the engine while the engine controller when the engines are turned on. (H1, 2, 3)

Causal Scenario 11.1a: The FCC does not request the desired power level to complete the commanded maneuver from engine controller because the FCC has a flawed process model. This could occur if:

1. The FCC is operating based on incorrect data about the environment due to incorrect or insufficient feedback and control inputs and is unable to form a message to request power from the engine.
2. The FCC isn’t aware that the engines have been powered on because the feedback from the engine controller has been lost.
3. The FCC loses power.

Example Requirements:

a. The engine controller shall have a default power level that allows the aircraft to be landed if it loses communication with the FCC.

UCA 11.2: The FCC requests the incorrect power level from the engine controller. (H1, 2, 3)

Causal Scenario 11.2a: The FCC requests the incorrect power level from the engine controller because it has a flawed process model of the aircraft and/or the operating environment. This could occur if:

6. The FCC is operating based on incorrect data about the environment due to incorrect or insufficient feedback and control inputs (e.g. incorrect sensor data). It passes incorrect data to the engine controller which results in the wrong power settings.
7. The FCC incorrectly believes that there is an engine fire and requests that the engine be shut down to extinguish the fire. Shutting down the engine results in insufficient power for the maneuver.
8. Data in the FCC is corrupted which results in incorrect information being passed to other aircraft systems that depend on the FCC.
9. The FCC receives an incorrect weight on wheels (WOW) indication and commands the engine controller to bring the engine to idle. This could occur if the WOW sensors malfunction or the aircraft inadvertently touches down on a surface that is not safe to land on.
10. The FCC has an incorrect engine model because the model was developed incorrectly or the engines are performing unexpectedly.

Example Requirements:

a. The FCC shall have multiple sources of feedback from the aircraft and environment that are checked to help ensure accurate data is used.
b. Before shutting down an engine, the FCC shall check to see if the aircraft is in an attitude that can be sustained without the power from the engine.
c. Checksums or other data quality control measures shall be used to protect against data corruption and its consequences for the system.
d. The WOW sensors shall not signal a mode change until TBD% of the aircraft's weight has been detected.