Extending the Human-Controller Methodology in Systems-Theoretic Process Analysis (STPA)

by

Cameron L. Thornberry

Bachelor of Science in Aerospace Engineering
United States Naval Academy, 2012

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

MASTER OF SCIENCE IN AERONAUTICS AND ASTRONAUTICS

AT THE

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

JUNE 2014

Signature of Author:_____
Department of Aeronautics and Astronautics
May 22, 2014

Certified by: _____
Nancy G. Leveson
Professor of Aeronautics and Astronautics and Engineering Systems
Thesis Supervisor

Accepted by: _____
Paulo C. Lozano
Associate Professor of Aeronautics and Astronautics
Chair, Graduate Program Committee

Extending the Human-Controller Methodology in Systems-Theoretic Process Analysis (STPA)

by

Cameron L. Thornberry

Submitted to the Department of Aeronautics and Astronautics on May 22, 2014 in Partial Fulfillment of the Requirements for the Degree of Master of Science in Aeronautics and Astronautics

ABSTRACT

Traditional hazard analysis techniques are grounded in reliability theory and analyze the human controller—if at all—in terms of estimated or calculated probabilities of failure. Characterizing sub-optimal human performance as "human error" offers limited explanation for accidents and is inadequate in improving the safety of human control in complex, automated systems such as today's aerospace systems. In an alternate approach founded on systems and control theory, Systems-Theoretic Process Analysis (STPA) is a hazard analysis technique that can be applied in order to derive causal factors related to human controllers within the context of the system and its design. The goal of this thesis was to extend the current human-controller analysis in STPA to benefit the investigation of more structured and detailed causal factors related to the human operator.

Leveraging principles from ecological psychology and basic cognitive models, two new causal-factor categories—*flawed detection and interpretation of feedback* and the *inappropriate affordance of action*—were added to the human-controller analysis in STPA for a total of five categories. In addition, three of the five human-controller causal-factor categories were explicitly re-framed around those environmental and system properties that affect the safety of a control action—the process states. Using a proposed airspace maneuver known as In-Trail Procedure, a former STPA analysis was extended using this updated human-controller analysis. The updated analysis generated additional causal factors under a new categorical structure and led to new instances of specific unsafe control actions that could occur based on additional human factors considerations. The process, organization, and detail reflected in the resultant causal factors of this new human-controller analysis ultimately enhance STPA's analysis of the human operator and propose a new methodology structured around process states that applies equally as well to an automated controller.

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

ACKNOWLEDGEMENTS

To Dr. Nancy Leveson, you not only shifted my entire perspective on systems thinking and safety engineering, but you encouraged me to cultivate my own interests and pursue my own problems. For that I am forever grateful.

To the lab, I sincerely thank you all for the conversations held in and around 33-407 that helped spark my own academic growth over the past two years. I want to especially thank you, Cody, for taking the time to help guide my journey and provide feedback on my early ideas. Dan, I also have you to thank for the deep discussions on human experience and our position (past, current, and future) given the inevitable rise of the machines.

To Dr. John Flach, Dr. Lawrence Hettinger, and Dr. Neville Stanton, a special thanks for sharing your wisdom and unique insights into the human operator. Talking with you all helped to provide the inspiration for this thesis.

To MIT Triathlon and Cycling, thanks for balancing the mental challenge with a physical one. This entire grad school experience would've undoubtedly been different had it not been for those frequent visits to the depths of the pain cave and all the fun along the way.

Finally, to my entire family, you provided the unshakeable foundation of love and support that made this all worthwhile. I love you all.

Table of Contents

List of Figures

List of Tables

1 Introduction

1.1 Background

As technology evolves, systems of greater complexity necessitate a more thorough understanding of system control. Not only has the digital revolution recast our day-to-day lives, from the clouds of internet storage to the walls of Facebook, this revolution has also punctuated the way complex systems like nuclear power plants and commercial aircraft, for example, are designed, built, and controlled. As a result, software and automation are now ubiquitous throughout the complex systems of our modern world. Not only this, but the dizzying pace of technological and digital advancement now asks humans to control unprecedented levels of automation that has shifted—and not necessarily decreased—the workload of their human controllers. As Dr. Sidney Dekker aptly points out in *The Field Guide to Understanding Human Error*, new technologies are enabling systems to be driven closer to their margins while humans are left to juggle entire patchworks of automation and competing high-level system goals and pressures in order to make the entire system function in actual practice [1]. The result: more accidents attributed to "human error."

While the term "human error" is a misnomer for deeper symptoms that plague a system, this language that ascribes human failure is still widely used in hazard analysis methods and safety investigations, and fundamentally undercuts our ability to understand this new coupling of humans and automation. The crash of Air France 447 in 2009, for example, highlighted just this. As the French Civil Aviation Safety Investigation Authority (BEA) concluded in its final report of the accident, the reasons for the crash center on the failure of the crew—whether they "provided inappropriate control inputs," "[identified their deviation from the flight path too late]," or "[failed] to diagnose the stall situation and consequently a lack of inputs that would have made it possible to recover from it [2]." While this 2012 report is incredibly damning to the human controllers through availability of hindsight bias, the real causes of this crash run much deeper and involve systemic factors like the automation onboard, the side-sticks used for control inputs, lack of and too much feedback in certain areas, etc. Another example pairing humans and automation in complex systems is the 2009 crash of Turkish Airlines Flight 1951 at the

Amsterdam Schiphol Airport. While the Dutch Safety Board's conclusions thankfully do not identify human error as a primary cause of this accident, it was stated that the combination of equipment failure, inadequately designed automation, and *poor crew resource management* led to the death of nine people and injury of 120 [3]. It is modern examples like these that showcase the fact that not only are human controllers subject to stress, fatigue, and similar traditional human factors, they are also vulnerable to new systemic factors involving software and automation with potentially deadly consequences. Unless hazard analysis techniques are better equipped to solve these new issues involving human operators at the helm of complex systems, accidents like these are still bound to occur.

While learning from accident investigations is one way to improve system safety, the second method of doing so involves analyzing the safety of a system during design or while it is in operation. Current approaches in this second method are fundamentally grounded in chain-of-event models in which it is assumed that unlikely scenarios occur in succession and pass through multiple barriers of defense to lead to an accident. Examples of analysis methods based on this model include Probabilistic Risk Assessments (PRA), Fault Trees, Event Trees, Failure Mode and Effects Analyses (FMEA), and Hazard and Operability Studies (HAZOP). These types of hazard analysis techniques are historically grounded in the simple electro-mechanical systems of the past where component reliability was synonymous with the safety of the entire system. Although heuristically reasonable, this mentality equates component reliability with safety, which, as will later be explained, is simply not true. Moreover, these models approach the human operator as just another component in the system and assign simple probabilities to "human error" or "human failure" in these causal scenarios. As such, it is not surprising that these traditional safety analyses still fail to uncover and prevent the accidents of today.

In an entirely new safety approach that looks far beyond linear causation, Systems-Theoretic Process Analysis (STPA) is a hazard analysis technique grounded in systems thinking where safety is treated as an *emergent property* of the system. STPA itself is part of the larger framework of Systems-Theoretic Accident Model and Processes (STAMP)—a new accident model developed by Dr. Nancy Leveson that is based on systems and control theory.

The application of systems thinking in STPA expands traditional safety analyses by coherently capturing nonlinear relationships between components and by treating safety as a *control problem*, instead of merely as a component reliability problem. Therefore, when dealing with software, automation, and overall system design, STPA is an ideal candidate for analyzing the safety of complex systems—more on this in the next chapter.

1.2 Motivation

Despite this recent advance in a system-theoretic approach safety, a problem that continues to plague all safety methods is the analysis of human controllers within complex systems. Some hazard analysis techniques do not even include an analysis of the human controller, but almost all of those that do, as alluded to earlier, view the human as another component in the system that is susceptible to failure. Identified accident scenarios that involve the human controller are traditionally assigned probabilities of "human failure" or "human error" as a recent example from 2008 shows in **Figure 1**.

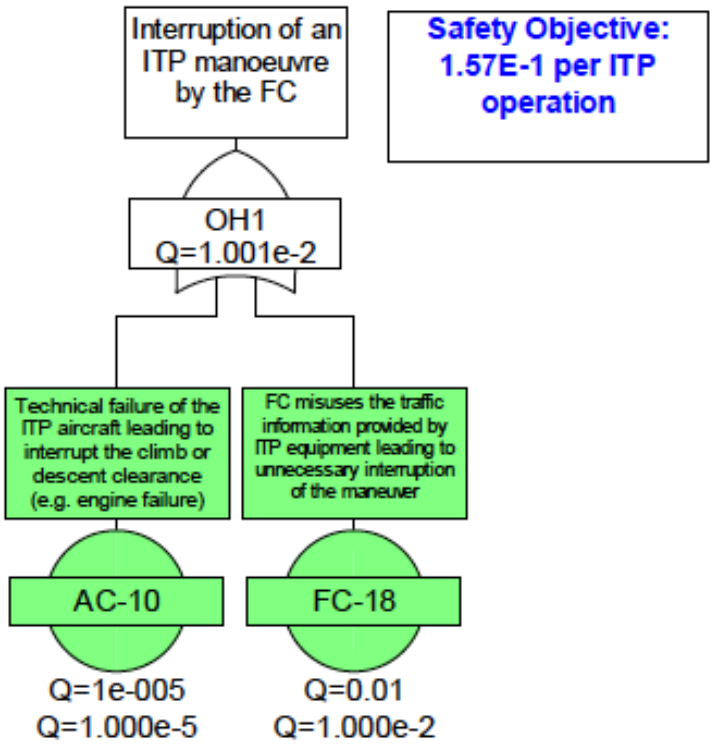

Figure 1. Example of Human Failure Probabilities [4]

In this fault tree example, the interruption of an In-Trail Procedure (ITP) maneuver by an aircraft flight crew (FC) is judged to have a probability of occurring once in every 100 flight hours, as denoted by the 1.001e-2. This answer is the result of combining two probabilities of failure, one of aircraft failure and the other on behalf of flight crew failure that was generated by "expert operational opinions" provided by various pilots and air traffic controllers [4]. While well-intentioned, the probabilities of human error in **Figure 1** try to simply quantify a surface-level phenomenon that is known as "human error," which, as will be discussed in Chapter 2, carries multiple flawed assumptions.

STPA, on the other hand, approaches the human controller through the lens of system safety. The overall goal of STPA, like any other hazard analysis method, is to generate hazardous scenarios that could lead to an accident. Through systems and control theory, in contrast to the limited scope of linear combinations of component failure probabilities like **Figure 1**, STPA is able to uncover significantly more scenarios and causal factors that could lead to an accident.

One of the greatest strengths in STPA is that it can detail causal factors related to controllers, software, component interactions, and overall system *design*. In addition to these systemic causal factors, the treatment of the controller (whether human or automated) also analyzes the impact that feedback and internal models of the system states have on the controller, and how flaws in these could lead to an accident. However, while STPA has come far in improving the analysis of automated and human controllers, there still remains room to improve the techniques of generating and categorizing causal factors related to the human controller and to integrate concepts from within the field of Human Factors.

1.3 Goals and Approach

In order to improve the clarity, organization, and analysis methodology for generating human-controller causal factors, this research seeks to accomplish one objective:

1. To extend the causal-factor methodology of the human controller in STPA.

This goal will be approached by first explaining the how current safety methods analyze the human controller in Section 2.1. In Section 2.2, the STPA process will be explained in greater detail before the larger domain of Human Factors is discussed in Section 2.3. Using human factors concepts from Chapter 2, the current human-controller model in STAMP theory will then be deconstructed and rebuilt in Section 3.1, before the STPA causal-factor analysis of the human controller itself is updated in Sections 3.2 and 3.3. An example involving a proposed airspace maneuver known as In-Trail Procedure will then be used in Chapter 4 to demonstrate the updated human-controller analysis developed in Chapter 3. Finally, Chapter 5 will summarize the updated human-controller analysis with respect to the research objective above as well as discuss avenues for future research regarding the human operator.

2 Literature Review

2.1 Safety Approaches to the Human Controller

All approaches to safety, whether through design, current analysis, or post-accident investigation of a system, are founded on some *accident causality model* of that system. Each model houses a set of theoretical assumptions that explain why accidents occur and how they can be prevented, and influence the way we conceptualize and abstract the mechanisms of accident causation [5]. These assumptions then necessarily affect how the human operator is treated and analyzed with respect to accident causation within each model—if at all. Traditional safety assumptions, which govern almost all present-day safety methods, stand in sharp contrast to those made in STAMP and therefore lead to definitively different conclusions about the human controller's role in accidents.

2.1.1 Traditional Safety Assumptions

Traditional safety analysis methods are those that rely on a linear chain-of-events model of how accidents occur. If human operators are involved in this chain of events these models are quick to point to human failure or "human error" as a primary cause, just as the investigators in the Air France 447 final report concluded. This method of thinking is not new, and in fact formally dates back to 1931 when Herbert Heinrich first introduced his Domino Accident Model, shown in **Figure 2**.

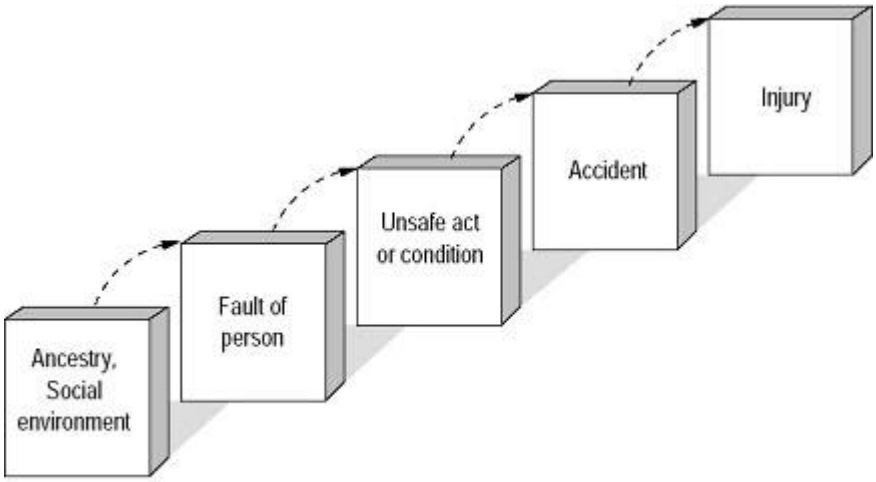

Figure 2. Heinrich's Domino Accident Model

The crux of Heinrich's model asserts that accidents occur through a linear propagation of events, much like a series of dominos, and that by removing any one of the events—or dominos—in the sequence, accidents could be prevented. The first two dominos in **Figure 2** cover flaws in the person involved in the accident, followed by the third domino of the actual unsafe act, the fourth domino of the resultant accident, and the final domino of related injuries. His model quickly took off because it intuitively made sense: we humans naturally process the world around us through a chain-of-events mentality, and this heuristic made for an easy understanding of accident causality. Heinrich was an early proponent of "human error" (which he dubbed "man failure"), and blamed much of the accidents, 88% in fact, on the lack of "enforcement of proper supervision [6]." As the first three dominos in **Figure 2** clearly show, "human error" was almost an inescapable cause in accidents. In this subtle nod to Hobbesian theory, Heinrich laid the foundations of this view on human error that would percolate all the way to today's mainstream understanding of accident causation.

It was Heinrich's Domino Accident Model that paved the way for Reason's Swiss Cheese Model, one of the most widely recognized models in use today [7]. In line with the linear chain-on-events mentality, the Swiss Cheese Model is founded on the view that accidents occur when

each protection barrier in the system fails and "lines up," akin to holes in slices of Swiss cheese represented in **Figure 3**.

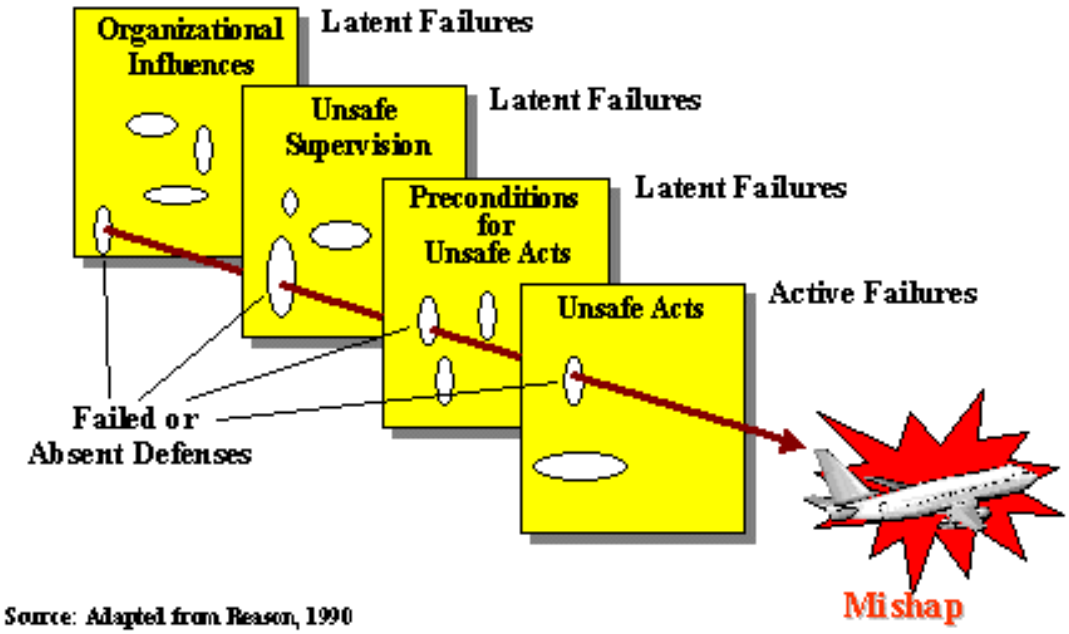

Figure 3. Reason's Swiss Cheese Model of Accidents

The two categories of failures in the Swiss Cheese Model in **Figure 3** are latent failures and active failures. Latent failures are defined as those decisions or actions that lie dormant in the system well before the accident sequence and are typically made by the "designers, high-level decision makers, managers, and maintenance staff," as represented by the first three slices of cheese [7]. Active failures, on the other hand, are those errors or failures that immediately lead to an accident in real time and are typically brought about by the operational controllers (e.g., human failure). All of these failures can then line up and when they do, an accident will occur through a linear sequence of events. Notice now that sources of human error are not just limited to the immediate human operator(s) in the proximate chain-of-events, as flawed human decisions and actions can now be traced back all the way to the controlling organization(s) involved in the accident.

Traditional linear models like Heinrich's Domino Accident Model and Reason's Swiss Cheese Model spawned almost all modern hazard analyses of the human controller, and with it, a set of implicitly unsound assumptions about the human operator. It should be noted that not all traditional hazard analysis techniques even account for the human controller, but those that do are strictly grounded in reliability theory. Examples include, but are not limited to, Probabilistic Risk Assessment (PRA), Cognitive Reliability Error Analysis Method (CREAM), Human Reliability Analysis (HRA), Fault Tree Analysis (FTA), Hazard and Operability Analysis (HAZOP), and Human Factors and Analysis Classification Systems (HFACS), all of which view the human as an independent component in the system. While each particular approach has its own nuances, the underlying assumptions about the human controller are based on faulty logic.

The first primary assumption in these approaches is that system safety is a product of component reliability. That is, if all of the components within a system do not fail, then the safety of that system is guaranteed. This, however, is simply not true. Safety is an emergent system property, whereas reliability is simply the absence of component failure. If all the components on an aircraft are reliable, for example, this will not prevent it from violating minimum separation standards or from colliding with other aircraft or the ground. Component reliability represents only one aspect of safety. The larger context, component interactions, and system design also have influence on this emergent property of safety. Furthermore, this assumption implies that humans are interchangeable components and their "reliability" is independent of the system, its design, and the local context. In the modern landscape of software and automation, system design is now closely coupled to the effectiveness of human control. Approaching the human controller as a decoupled component of the system does little to improve system design and is inadequate for system safety.

The second assumption in these traditional hazard analysis techniques is that human reliability is a calculable or estimable fact. Since these methods date back to simpler electro-mechanical systems, failure rates of mechanical components were used to support the first assumption. These mechanical failure rates were and still are a known entity, much like the failure rates of wing spars or engines, for example. This same logic was applied to the human component to generate probabilities or qualitative estimations on the occurrence of sub-optimal

human decisions or actions, which are typically calculated by using previous human performance data, models, human factors experts, practitioner opinion, and methods of task analysis [8].

Despite all the effort put into these calculations, the human operators in direct control of complex systems do not randomly fail. This logic of human unreliability first implies that there is a deviation from correct, nominal, or prescribed behavior, which is heavily subject to hindsight and outcome bias [9]. Human unreliability is also labeled without the regard to the local context and what makes sense to the operator at the time, something that many human factors experts now agree plays an immense role.* This assumption does not account for the immediate task environment that the human controller operates in and completely removes the human operator from their environment [10]. Furthermore, there is also no way to factor in the mental models of the environmental or system states that the human operator believes to be true. Much like a thermostat has an internal model of the immediate surrounding temperature, so too do humans have their own models of how the system is operating (i.e., what "state" it is in), and what the state of the environment is. Hopefully the human operator's mental models align with the reality of the system and environment, as the congruence of the operator's beliefs and the actual reality of the system and environment are absolutely necessary for system safety, an idea that will be discussed in more detail in the next section. Given the limitless rise of technology in complex systems today, human operators are more prone than ever to phenomena like automation surprise and mode confusion, yet these approaches to calculating human error cannot account for nor convey these issues regarding the human controller's mental models.

Also inherent in these assumptions is that the system would be reliable and safe if it weren't for the unreliable human controller. Assuming the human controller is first involved in the chain of events, the inherent logic of direct causality implies that if the humans did not "fail" then the entire system would not have failed. This, again, is not true. As STAMP theory will demonstrate in the next section, accidents involve the whole socio-technical system and not a subjective, causal chain of events. Moreover, human control is not a series of discrete control actions, but rather a combination of parallel, continuous control tasks. Removing this context of

* To include the views of Dekker, Flach, Rasmussen, Vicente, Klein, and Gibson.

control as well as often-competing high-level goals and the overall system environment will seriously limit the ensuing analysis.

Finally, labeling "human error," whether through quantitative or qualitative assessment, does little to actually improve system design. While the certainty expressed in human reliability may assure safety engineers, the impact that these probabilities and estimations have on system design is little to none. By the time this system has been tirelessly analyzed through traditional hazard analysis techniques, the system design has largely been fixed. Given real-world temporal and financial constraints, completely re-designing or overhauling the system is extremely costly and the more likely change will come in the reliability analysis itself. Most critically, the system designers have very limited information from these probabilities to negate "human error" and improve the effectiveness of human control. As Dekker puts it, "the classification of errors is not analysis of errors," especially when it relates to the human controller [10]. Instead of merely looking at what is judged as phenotypical, this manifestation of "human error" must be investigated and analyzed under a completely different paradigm in order to uncover the genotypical systemic factors.

2.1.2 System Safety Assumptions

System safety as defined in STAMP (Systems-Theoretic Accident and Process Analysis) is an entirely new accident causality model that uses systems and control theory to treat safety as an *emergent property* of a system. Since safety is viewed as a property based on the interactions of components with each other and the environment, STAMP leverages the systems theory concepts of emergence, hierarchy, and control to impose safety constraints throughout the system to enforce the overall system property of safety. In contrast to viewing accidents as a linear propagation of events that stem from an initiating cause, accidents in STAMP arise due to the inadequate control and enforcement of safety-related constraints on the development, design, and operation of the system [5]. Safety is therefore a control problem for the entire socio-technical system, an example of which is seen in **Figure 4**.

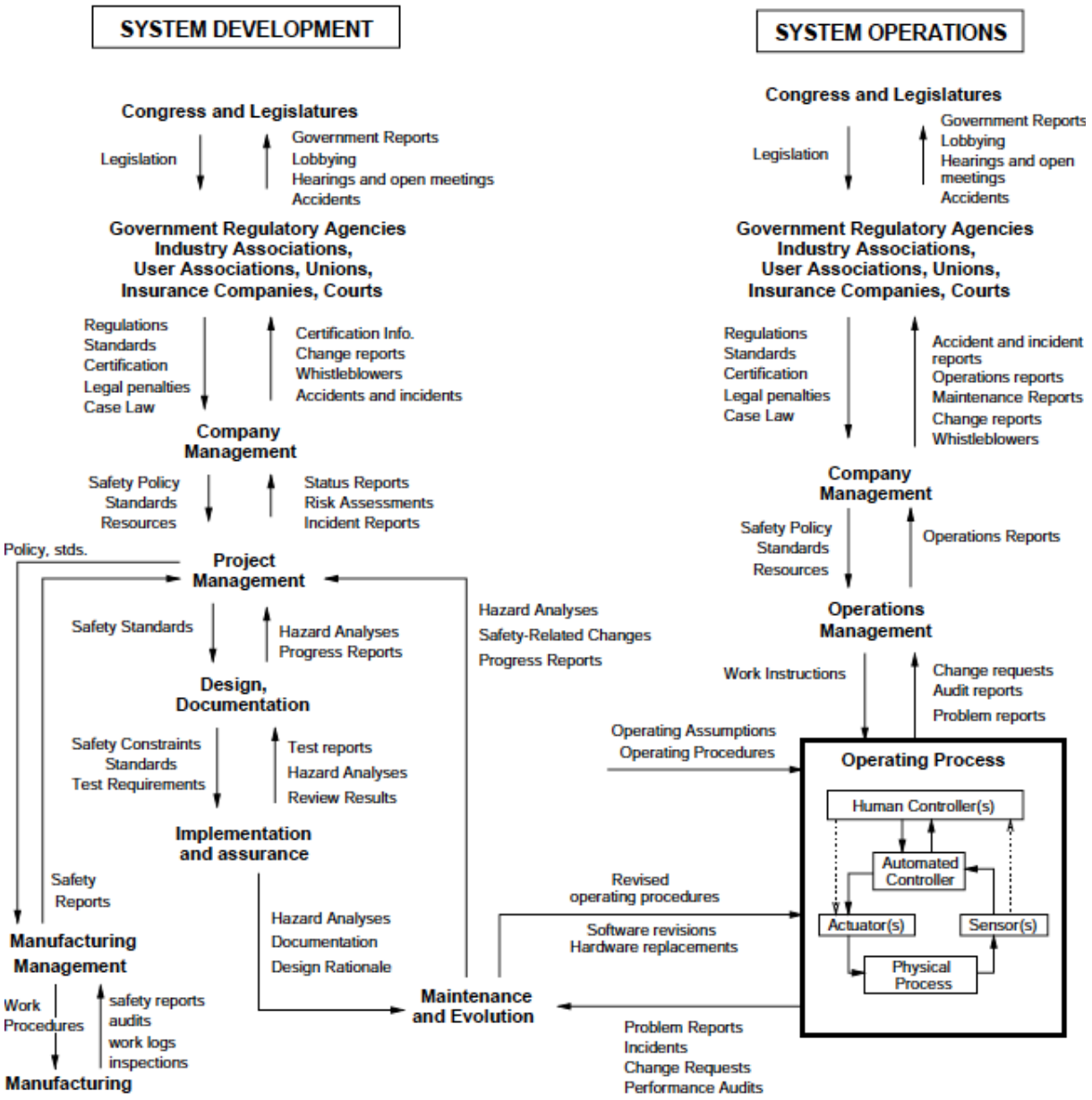

Figure 4. Example Hierarchical Control Structure [5]

In order to control for safety, STAMP utilizes safety constraints, control actions, feedback, and process models as the hierarchy in **Figure 4** shows. Each level in the hierarchy—whether physical, social, or human—can be treated like a controller that is responsible for controlling enforcing safety constraints on the level below it. As control theory necessitates, controllers need four conditions to control a process: a *goal* (safety constraint), the *action condition* (control

action), the *observability condition* (feedback), and the *model condition* (process model). The process model is the controller's model about the controlled process and the current state of the system and environment, much like a simple thermostat has its own model of the proximate ambient temperature. This example only considers one process model variable of temperature, whereas in reality both human and automated controllers have to deal with multiple process model variables during operation (e.g., an airline pilot flying a commercial aircraft).

The overall assumptions in STAMP, including those of the human controller, are summarized and contrasted with traditional safety assumptions in **Table 1**.

Table 1. System Safety Assumptions and Comparison [5]

Old Assumption	New Assumption
Safety is increased by increasing system or component reliability; if components do not fail, then accidents will not occur.	High reliability is neither necessary nor sufficient for system safety.
Accidents are caused by chains of directly related events. We can understand accidents and assess risk by looking at the chains of events leading to the loss.	Accidents are complex processes involving the entire sociotechnical system. Traditional event-chain models cannot describe this process adequately.
Probabilistic risk analysis based on event chains is the best way to assess and communicate safety and risk information.	Risk and safety may be best understood and communicated in ways other than probabilistic risk analysis.
Most accidents are caused by operator error. Rewarding safe behavior and punishing unsafe behavior will eliminate or reduce accidents significantly.	**Operator error is a product of the environment in which it occurs. To reduce operator "error" we must change the environment in which the operator works.**
Highly reliable software is safe.	Highly reliable software is not necessarily safe. Increasing software reliability will have only minimal impact on safety.
Major accidents occur from the chance simultaneous occurrence of random events.	Systems will tend to migrate toward states of higher risk. Such migration is predictable and can be prevented by appropriate system design or detected during operations using leading indicators of increasing risk.
Assigning blame is necessary to learn from and prevent accidents or incidents.	Blame is the enemy of safety. Focus should be on understanding how the system behavior as a whole contributed to the loss and not on who or what to blame for it.

As stressed in the fourth assumption in **Table 1**, "human error" no longer suffices as an explanation and is now recast through a system safety lens that considers the task environment and context surrounding the human operator. Since safety is treated as a control problem, STAMP investigates how safety constraints around the human controller could be violated through inappropriate control actions, feedback, and process models (hereafter referred to as "mental models" for the human controller) of the controlled process. In line with Dekker's

thinking, this assumption understands that humans often compete under high-level goals and pressures other than safety (e.g., time and money) and with a gamut of new technologies [1]. In another sense, this systemic approach is rooted in an "eco-logic" that considers the dynamic coupling between intelligent agents (the human controller) and the functional demands of the work environment (the system and environment), instead of merely approaching the human as an interchangeable component that can be analyzed independently of the system and context [11]. Section 2.3 will expand upon more of this ecological view of human control. As a last note, it is also assumed in STAMP that the human controller *wants to succeed* and avoid an accident, and therefore STAMP does not consider evil or malevolent motivations on behalf of the human controller.

2.2 How STPA works*

STPA (Systems-Theoretic Process Analysis) is a hazard analysis technique based on the STAMP accident causality model. The objective of STPA is to identify scenarios of inadequate control that could potentially lead to an accident, from which safety constraints can be generated and applied from system conception to operation.

2.2.1 The Process

STPA is an iterative process that consists of two primary steps. Built upon the System Engineering Foundation, STPA Step 1 identifies hazardous control actions that could lead to an accident, and STPA Step 2 determines how they can occur. The System Engineering Foundation is common across all hazard analysis techniques and starts by defining the accidents and high-level hazards. In the STAMP model, **accidents** are defined as an "undesired or unplanned event that results in a loss, including loss of human life or injury, property damage, environmental pollution, mission loss, etc.," and the **hazards** are "a system state or set of conditions, that, together with a particular set of worst-case environmental conditions, will lead to an accident [5]." These high-level hazards are then turned into high-level safety constraints that the designers and engineers use in their system engineering processes, which are akin to design requirements.

* Refer to [5] or [12] for more detail.

Then, in a final step that is unique to STPA efforts, a safety control structure is generated through a hierarchical, functional control diagram much like the example in **Figure 4**. Depending on the system the components may be human, physical, software, or organizational.

Once the entire System Engineering Foundation is established, the identification of hazardous control actions begins in STPA Step 1. Since hazardous system states arise from inadequate control, Step 1 investigates how inadequate control can be realized. The four types of unsafe control are [12]:

1. A control action for safety is not provided
2. Providing a control action creates a hazard
3. A potentially safe control action is provided too late, too soon, or out of sequence
4. A safe control action is stopped too soon or applied too long (for a continuous or non-discrete control action)

A fifth possibility—that a required control action is provided but not followed—will be addressed in STPA Step 2. These unsafe control actions, or UCA's, are generally categorized in a basic table like in **Table 2**, but are by no means restricted to this format.

Table 2. General Format for UCA Identification in STPA Step 1

Control Action	Not providing causes hazard	Providing causes hazard	Too early/too late, wrong order causes hazard	Stopping too soon/applying too long causes hazard
…				

After the unsafe control actions are identified, STPA Step 2 determines the potential causes of these UCA's. Using a basic control loop, each element of the control loop is examined to see how each UCA could be caused, detailed in **Figure 5**.

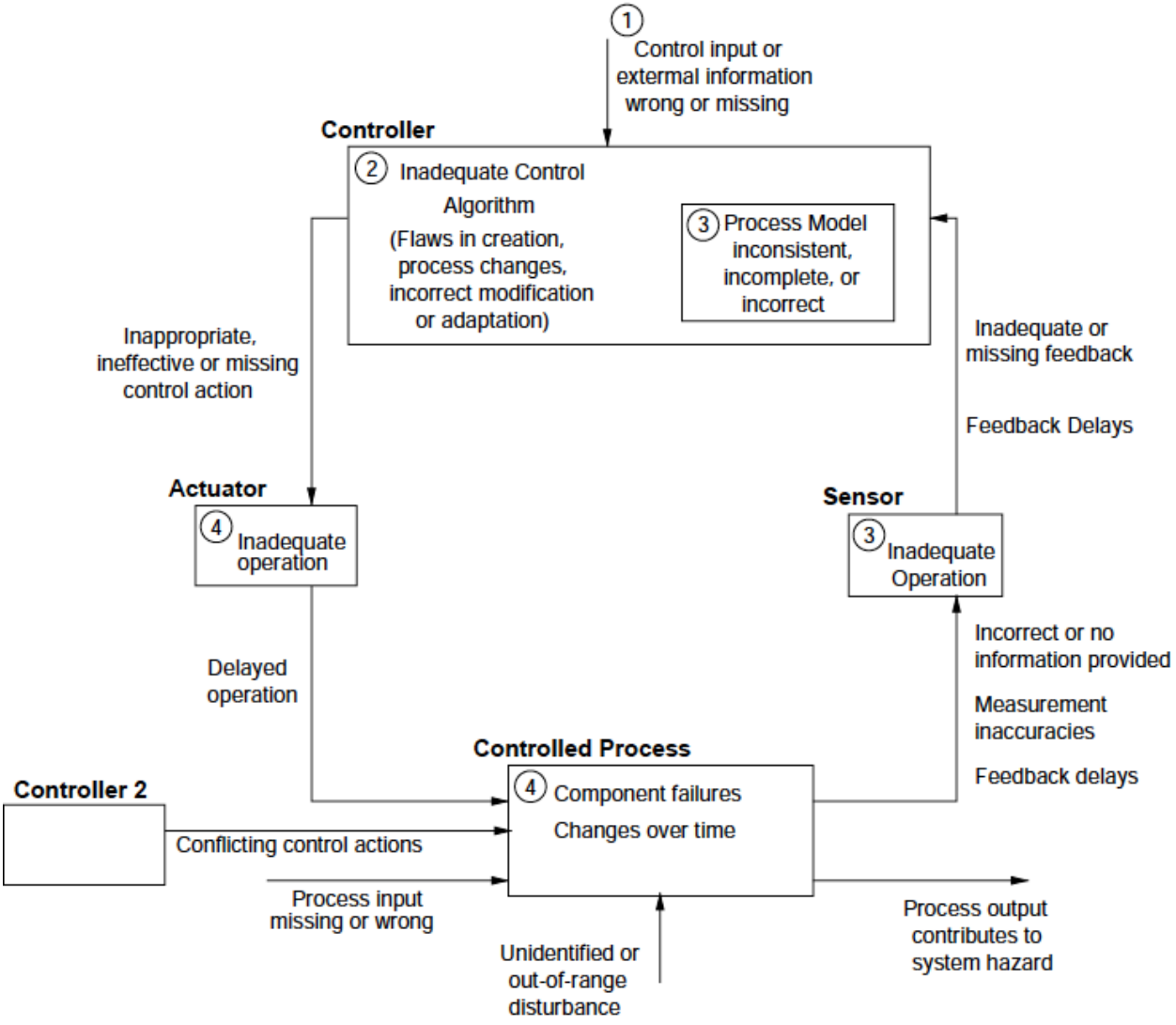

Figure 5. STPA Step 2 Causal Factor Examination [5]

As **Figure 5** shows, all elements and communication links are inspected to see how they could cause unsafe control, including the controller, whether human or automated. Once the causal factors and scenarios are generated using this template, STPA Step 2 concludes by considering the degradation in controls over time. These causal factors that identify scenarios of accident causation can then be refined into lower-level safety constraints that inform system designers to design against or mitigate these identified flaws. An important distinction here is that STPA is a

hazard analysis technique that is used to *inform* design, and not actually design the system itself—a particularly critical distinction that will surface when analyzing the human controller.

2.2.2 Process States and Process Model Variables

Within the STAMP model and hence STPA lies an important concept of process states. **Process states** are those real environmental and system states that necessarily affect the safety of a control action. Controllers, both human and automated, have their own models of these real process states, known as the **process model variables**. The difference between these two is crucial: process states exist in reality, whereas process models exist in the controller's mind or software. Accurate matching between reality and the controller's model of reality is essential for system safety, else it could lead to unsafe control. For a controller, the incongruence between process states and process model variables will, when paired with a particular set of worst-case environmental conditions, lead to an accident.

It is important to note that process states and process model variables are distinctly different than feedback. Simply stated, feedback is the channel that connects the system and environmental states of reality into the controller's model of those states. Any flaws in the feedback or in its presentation to a human controller may have immediate impact on the controller's process models, and thus, the safety of the system. This connection between process states, process model variables, and feedback will be explored in greater detail in Chapter 3.

2.2.3 Context Tables

As previously mentioned, there is no specified format for the identification and categorization of unsafe control actions in STPA Step 1. Unsafe control actions can therefore be couched in terms of the process states that affect the safety of that control action, as pioneered in the systematic method by Dr. John Thomas. In this approach, each unsafe control action is defined by four elements: the source (controller), the type (control action provided or not provided), the control action (the actual command that was provided or not provided), and the context (the process state that makes the action unsafe) [13]. There is no formula in identifying the process states, but as Thomas describes, the process states "can be derived from the system

hazards, from the required feedback in the control structure, and from other knowledge of the environmental and system states [13].” The general table for the systematic method, known as a **context table**, is shown in **Table 3**.

Table 3. The Systematic Format if Control Action Provided

Control action	Process State 1	Process State 2	Process State N	Hazardous control action?		
				If provided any time in this context	If provided too early in this context	If provided too late in this context
… provided						

Note that the control action in **Table 3** is labeled as “provided.” If the control action is “not provided,” the general context table would like it does in **Table 4**.

Table 4. The Systematic Format if Control Action Not Provided

Control action	Process State 1	Process State 2	Process State N	Hazardous if control action not provided?
… not provided				

In these tables, the process states can have two or more values to them that affect the safety of the control action. For example, the process state of an “emergency” onboard a simple subway train may either “exist” or “not exist.” Furthermore, since STPA leverages abstraction and hierarchy to deal with complexity, the initial high-level process state may, depending on the nature of the state, be broken down into lower-level process states. Continuing the example of “emergency,” this high-level process state might be composed of three lower-level process states of “smoke present,” “fire present,” and “toxic gas present [13].” Thus, the presence of smoke, fire, or toxic gas onboard a subway train would indicate the existence of an overall emergency state.

2.2.4 Human-Controller Analysis in STPA

When analyzing the potential for unsafe control on behalf of the human controller, STPA looks far beyond the surface level classification of "human error." As **Figure 5** showed, the causal factors that lead to accident scenarios now include the context of missing or flawed feedback, process model inconsistencies with reality, and inadequate control algorithms—a significant improvement over trying to calculate or estimate human reliability. In an effort to extend STPA's analysis of the human controller even further, however, more detailed models of and approaches to human control were pursued in the domain of Human Factors.

2.3 Approaches to Human Control in Complex Systems

2.3.1 The Domain of Human Factors

Transitioning from the previous discussion of hazard analysis techniques and the treatment of the human controller, the focus will now shift to the broad area of research of the human known as "Human Factors." As defined in this thesis, Human Factors (capitalized) is an amalgamation of the many sub-sets of investigation into the human that generally revolve around the categories of:

- Ergonomics
- Psychology
- Humans and automation
- Human sociology
- Human performance

Although these categories are semantically separate, much of the research in this domain branches across multiple categories and disciplines. In the immediate effort of incorporating more concepts from Human Factors into STPA, however, two specific areas of research were explored: the relationships between the human controller and system ecology (ecological situation) and the basic modeling of human cognition and decision-making (cognitive modeling).

2.3.2 Ecological Situation

Just as all hazard analysis techniques rely on some underlying accident causality model, all approaches to Human Factors are based on some philosophical framework that describe the fundamental nature of reality. As Dr. John Flach eloquently summarizes in *What Matters (forthcoming)*, there are four unique ontological positions, three of which are founded on the diagram in **Figure 6** [14].

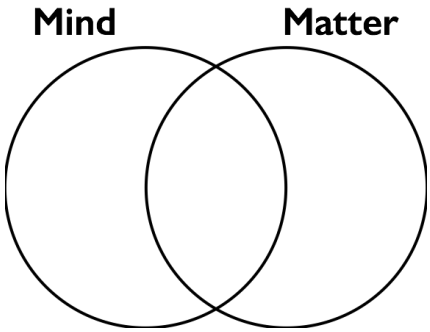

Figure 6. Mind or Matter? [14]

Materialism: The ontological view that reality is fundamentally composed of matter. All answers to and explanations of our questions reside in the physical properties of matter (e.g., in the chemical and electrical properties of our brain).

Idealism: The assumption that reality is based exclusively in the mind and that our everyday experiences only become "real" through the mind.

Dualism: The prevailing ontology that the reality of the mind is separate from matter, and that the reality of matter is separate from the mind. Thus, two different sciences are needed: one for the mind (psychology) and one for matter (physics). Linking the two, however, has always posed a problem.

In his philosophical musings in *Essays in Radical Empiricism (circa 1912)*, psychologist and philosopher William James was the first to propose a fourth ontology that treated the human mind and its surrounding environment as one experience—a view that laid the theoretical

foundations of what is known today as ecological psychology. As James writes, "the peculiarity of our experiences, that they not only are, but are known, which their 'conscious' quality is invoked to explain, is better explained by their relations—these relations themselves being experiences—to one another [15]." Through his writings, William James challenged the conventional approaches to mind and matter by creating a new view of reality as depicted in **Figure 7**.

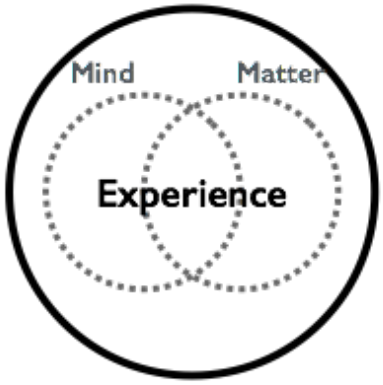

Figure 7. Mind and Matter as a Single Experience [14]

Radical Empiricism: The ontological view that there is only one reality—experience. Our perceptions are emergent properties that reflect constraints associated with both mind and matter.

At the vanguard of this fourth ontology is a new science known as ecological psychology that explores the full range of human experience through both mind and matter. This new mode of thinking integrates especially well into systems theory (and vice versa) when considering a human controller, and one of the greatest strengths in this new science is the way both feedback and human control are analyzed.

This ecological approach to feedback design is unique in that it is focused on channeling the deep structural constraints or "state" variables of the work ecology (task environment) in an effective way to the human controller. Whereas conventional approaches to feedback design (e.g., technology-, user-, or control-centered approaches) define the system in terms of the human and the machine, this ecological approach defines the system relative to its function in the

larger work ecology in order to design *meaningful* feedback [11]. A driver's knowledge of their speed means little until it is put in the context of their task objective (e.g., driving to work or racing Formula 1) and their vehicle capabilities (e.g., Ferrari or minivan), for example. The states of going "too fast" or "too slow" are therefore a property of *meaning* (i.e., what matters) that can only be understood relative to the ecology or situational context. Meaning, this is to say, is an emergent property of the system. The purpose in ecological interface design is to effectively align the "state" variables and functional constraints that matter in the work ecology with the human controller's own model of those variables through feedback. This ecological approach to framing feedback around the "state" variables has direct application to STPA principles and will be explored further in Chapter 3.

Another concept unique to ecological psychology is the idea of *affordance*. Although commonly thought of as a property of the environment (matter), **affordance** is defined as "the coupling of motor effectivities (mind, or at least agent based) and the opportunities in the ecology (matter)," and is what allows the human controller to change the state variables in the work ecology [14]. An example of affordance would be the combination of human arms and hands (motor effectivities) and the handlebars on a typical bicycle (opportunity in the ecology) that allow a bicycle to be steered. Neither the human's arms and hands nor the handlebars afford the steering of the bicycle on their own, but the coupling of the human's arms and hands *and* the handlebars allow for control over bicycle's yaw orientation (assuming just one axis in this simple example). This concept of affordance also has direct applicability to STPA and will be pursued in Chapter 3.

2.3.3 Cognitive Modeling

Although human cognition, information processing, and control is best understood relative to a situation or work ecology, there do exist some basic concepts of human psychological mechanisms that have potential crossover to STPA. There are countless cognitive models in the literature, and no intent was made to be all-inclusive given the criticism that the majority of these models analyze the mind separate from matter and the overall work ecology. However, two well-known cognitive models will be explored to illustrate some of the basic

psychological mechanisms involved in information processing and decision-making: the OODA Loop and the Skills, Rules, and Knowledge (SRK) Model.

One of the most acclaimed models in basic human cognition was proposed by Air Force Colonel John Boyd and classically known as the OODA (Observe-Orient-Decide-Act) Loop, represented in **Figure 8**.

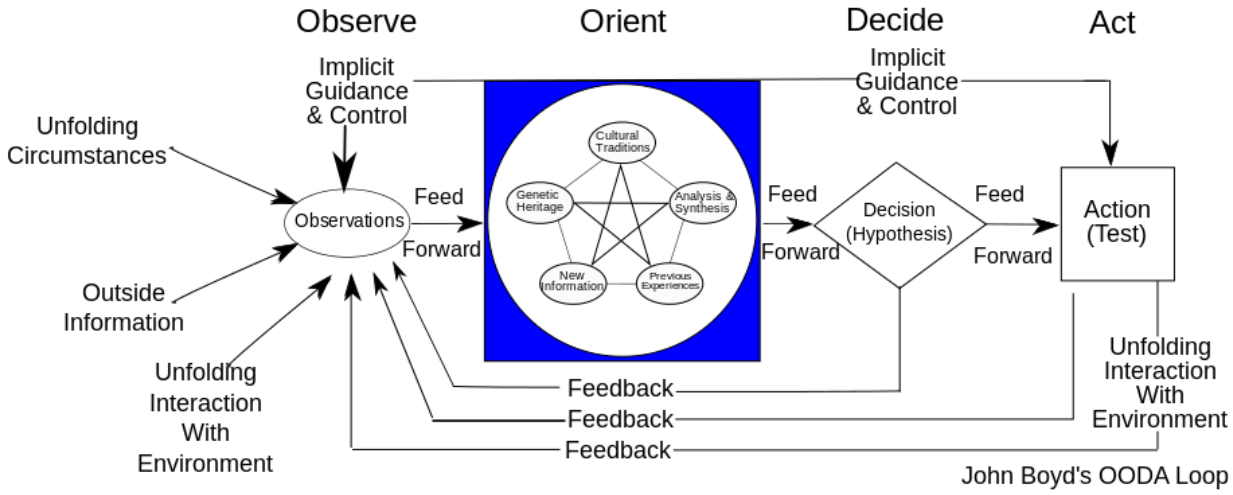

Figure 8. Boyd's OODA Loop [16]

This model was initially developed to frame and explain both military tactical-level and strategic-level operations, but the ready applicability to other industries and the Human Factors domain quickly made this model popular [17]. As the name suggests, this model has four main components to it: observation, orientation, decision, and action. When performing a task, the conditions and states necessary for the successful execution of that task are first observed. Following this observation, the controller then orients themself to the problem space (work ecology) before deciding upon a course of action and eventually realizing that action. This process is not a set of psychological sequences, but rather a continuous process much like a control loop. One of the greatest strengths to Boyd's model, aside from its simplicity, is that it can be applied across all domains, time scales, as well as to different controllers, whether human, automated, or organizational. The flexibility in Boyd's approach will see basic application to STPA in Chapter 3.

Shifting focus to the second model, Dr. Jens Rasmussen pioneered another approach to human cognition in his Skills, Rules, and Knowledge (SRK) Framework as depicted by his model in **Figure 9**.

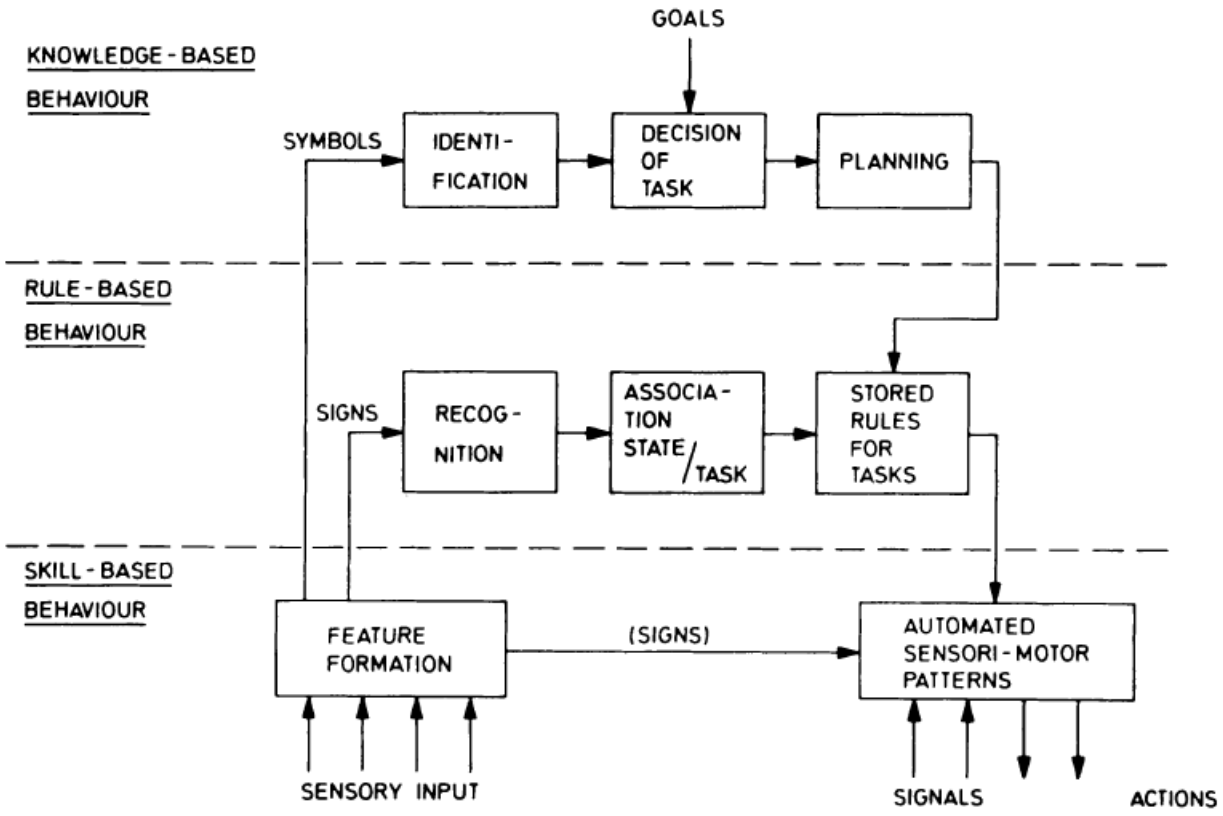

Figure 9. The Skills, Rules, and Knowledge Model [18]

Rasmussen's SRK Model is broken down into three separate levels of performance: skill-, rule-, and knowledge-based behavior. Skill-based behavior represents those lower-level sensory-motor skills that take place without conscious thought, like riding a unicycle or dribbling a basketball, for example. Rule-based behavior considers performance that is conducted through the use of previously stored rules within the mind, such as cooking a family recipe, for example, and usually occurs under an implicitly formulated goal. The last category of knowledge-based behavior considers human problem solving with unfamiliar tasks and limited or no rules for

control—but with an explicit goal or objective in mind. Learning to play chess for the first time is an ideal example of this.

Of particular interest in Rasmussen's SRK Framework is the way in which he categories feedback for these three behaviors. Skill-based behaviors are fed by a source of feedback known as **signals**—those "continuous quantitative indicators of the space-time behavior of the environment [18]." The next level of feedback used to make rule-based decisions is known as a **sign**—physical states or situations in the environment that inform rule-based behavior. Lastly, the categorization of feedback that supports knowledge-based behavior is known as a **symbol**—a functional property in the environment that is generated through human meaning. This idea of meaning, in fact, directly relates back to the previous discussion on the role of feedback in ecological interface design. Given the overall goal of extending the human-controller analysis in STPA, concepts from both ecological psychology and basic cognitive modeling, to include the SRK Framework, will be applied to the human controller in the next chapter.

3 STPA and the Human Controller

Compared with traditional hazard analysis techniques that stop after assigning human reliability, STPA applies systems thinking to derive both human and automated controller causal factors that relate to flawed feedback, inconsistent process models, and inadequate control algorithms. Using concepts explored in cognitive modeling and ecological situation, this chapter will first offer updates to the human-controller model in STAMP theory in Section 3.1. An update to the human-controller causal-factor analysis in STPA Step 2 will then be pursued in Section 3.2 before the application of this updated human-controller causal-factor methodology is detailed in Section 3.3.

3.1 Updating the Human-Controller Model

Figure 10 showcases the general STAMP model for a human controller coordinating with an automated controller to control some physical process. This diagram is a general model—one that can be tailored to analyze any system in STPA—but of particular interest in **Figure 10** is the human controller, outlined in the red dashed box.

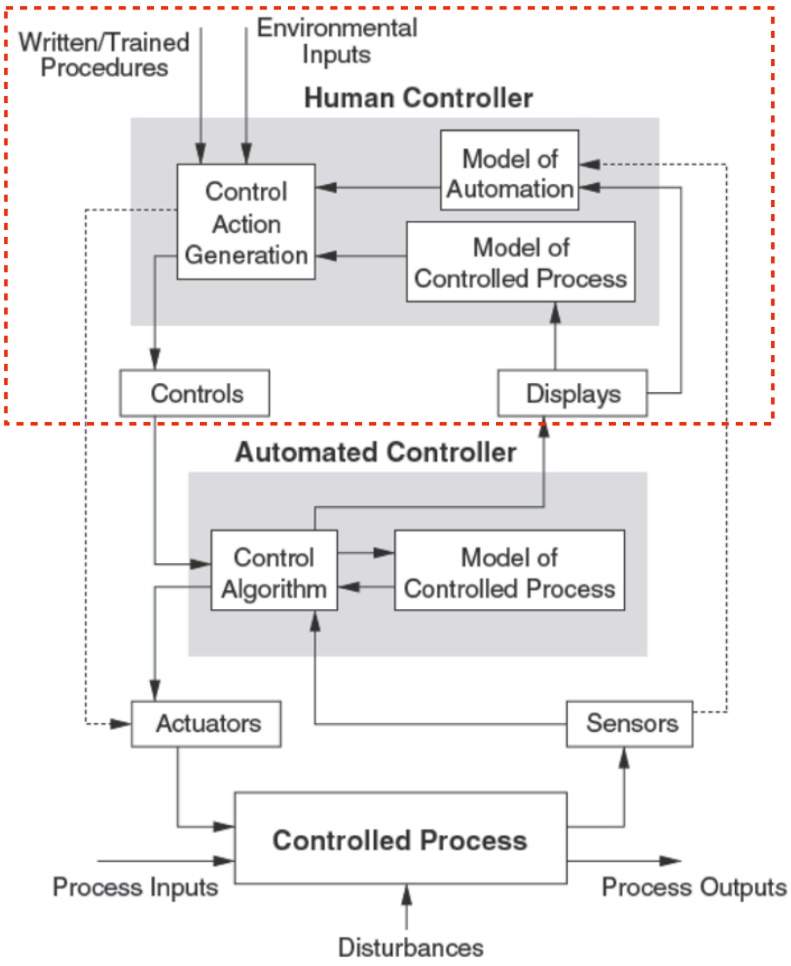

Figure 10. The Current Human-Controller Model [5]

Starting in the bottom right-hand side of **Figure 10**, the displays from the automation and the sensors from the controlled process connect to the human controller through any mode of feedback (e.g., visual, auditory, tactile) as designed in the system. The feedback from the displays and sensors then updates the human's mental models through either direct or indirect access as indicated by the solid and dashed lines, respectively. In this diagram, there are two basic mental models the human is aware of: the model of the automation and the model of the controlled physical process itself. These mental models are then used in dynamic (vice static) human control action generation that operates under the influence of written/trained procedures and environmental inputs. Once a control action has been generated, the control action is sent to the automation or the controlled physical process, through controls or actuators, as either a

primary or backup control action, as indicated by the solid and dashed lines, respectively. When zooming out in **Figure 10** and contrasting the red box of the human controller with the automated controller directly beneath it, note that there is little distinction between the two. The only change to the human controller is an unseen dynamic control algorithm (vice a static, automated algorithm) and an additional process model of the automation itself. Otherwise, the two controllers are identical.

In response to this, the model of the human controller can be updated by using the Human Factors concepts explored from Chapter 2 to better reflect traits unique to the human operator. The updates are seen in **Figure 11**.

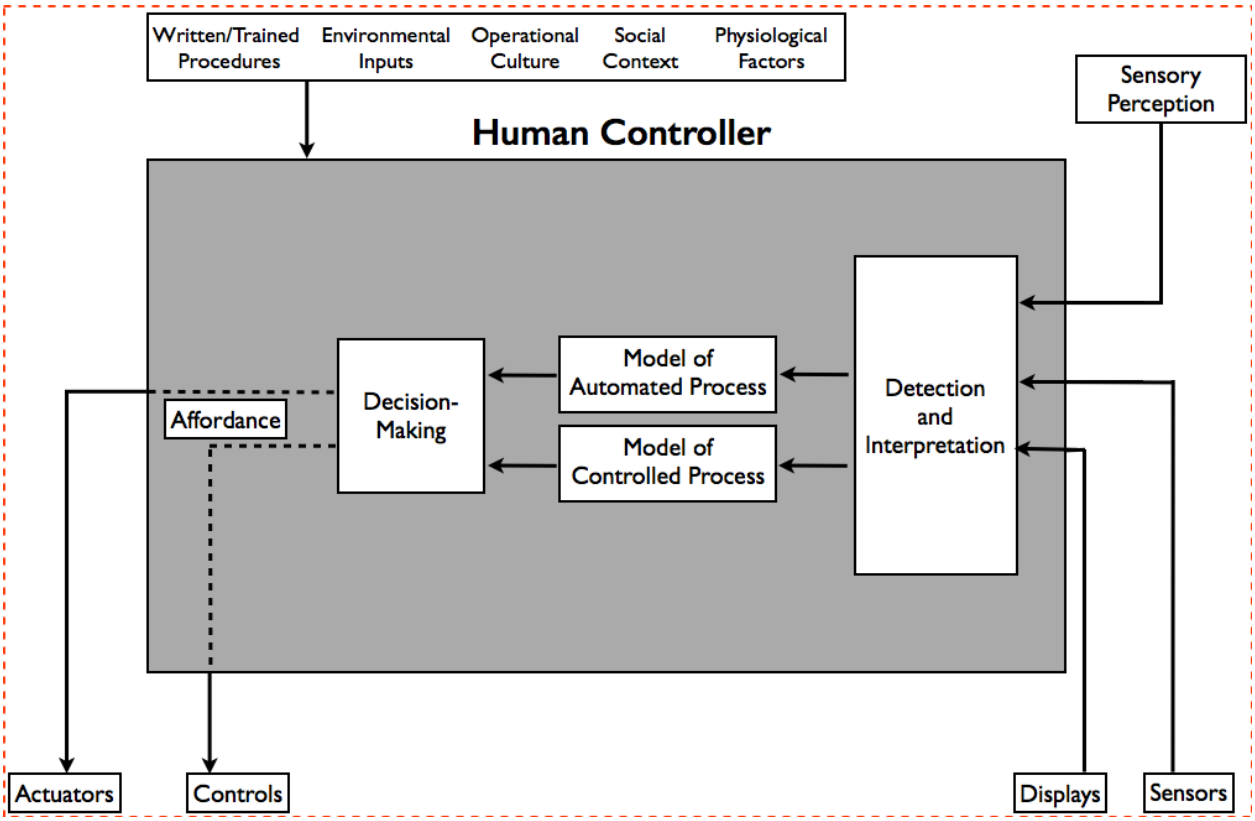

Figure 11. The Updated Human-Controller Model

Working from the bottom right-hand corner of **Figure 11**, the displays from the automation and the sensors from the controlled process connect to the human controller just as they did in **Figure 10**. In addition to these two sources is a clarified tertiary source of the human operator's own

sensory perception that was refined from the original "environmental inputs" in **Figure 10**. This sensory feedback includes that raw visual, proprioceptive, and vestibular feedback the human receives *that has not been designed into the system*, similar to Rasmussen's "signals." The inertial accelerations felt by a human while driving a car or flying an airplane would be an example of this type of feedback.

All three of these feedback categories then feed into a "detection and interpretation" stage through which the mental models are updated. Most generally agreed upon models of human cognition, like the OODA loop and SRK Framework, include some stage of human information recognition or observation, and thus this new category was added to **Figure 11**. The human operator must accurately process and understand the feedback from the displays, sensors, and their own sensory perception in order to form the correct mental models. It is important to note that *accurate* human interpretation of the feedback in **Figure 11** is defined to be *as the process states exist in reality*. This relates to ecological interface design where the intention is to uncover the situated "state" variables in the ecology and relate them to the human controller through feedback. Therefore, if the process states in STPA that are channeled through feedback are interpreted in any way other than as reflected in reality, the mental models of the human may not align and cause an accident—an allusion to a causal factor that will be explored in the next section.

Following the detection and interpretation of these feedback channels, the mental models of the human operator are then generated or refreshed, just as they were in the original model in **Figure 10**. From these mental models comes the renamed "decision-making" process for control actions. During operation, the human operator must consider and decide on higher-level goals within the overall control-action decision process. Not only do these higher-level decisions affect lower-level control actions, but these decisions can also transpire across different time scales as well, from seconds to hours—a matter of complexity that for now will be summarized as "decision-making."

Once a control action has been decided upon, the human operator must then afford the action before it can be realized. Although strictly thought of as a property of the environment, affordance (once again) is defined as the coupling between human motor effectivities and the

opportunities in the ecology of the system [14]. If the human operator cannot afford an action in time because of a poorly designed interface, for example, or if the human affords a control action unknowingly (e.g., automation surprise), an accident *will* occur when paired with a worst-case set of environmental conditions—more on this idea in the next section.

Once the control action is realized it is then sent to the actuators and/or controls just as in the current human-controller model of **Figure 10**. The entirety of this process, from detection to control action, takes place under the influence of a variety of external factors as seen in the top box in **Figure 11**. These include written/trained procedures, other external environmental inputs not related to lower-level sensory perception, an operational culture, a social context, and physiological factors (e.g., stress, fatigue, workload). All of these factors affect how the human controller detects and interprets feedback (e.g. through trained scan patterns) [19], how they assimilate feedback into accurate mental model representations, and how they make decisions accordingly. This is to say that every stage within the entire human operator "box" is affected by these factors and that they must be considered when analyzing the human controller.

Altogether, this updated model discretizes the abstract cognitive mechanisms and affordance component of the human operator when controlling a general system. As seen in **Figure 11**, these cognitive processes have been refined into the categories of "detection and interpretation," "mental models," and "decision-making" with the addition of "affordance"— the precursor to any control action. These categories are not comprehensive, however, as there are near limitless ways to model human cognition and information processing in the domain of Human Factors. Any one of the cognition components in **Figure 11** could also be elaborated further by culling more detail from cognition models and theories. As Dekker warns, however, these detailed models are impossible to prove or disprove [1], and insofar as improving STPA Step 2 human controller hazard analyses, a simple yet precise picture of human cognition is all that is needed. In pursuit of extending the human-controller methodology in Step 2, this updated model was applied to the causal-factor analysis itself in Section 3.2.

3.2 Updating the Human-Controller Analysis

The original controller causal factors analysis outlined in the red dashed box in **Figure 12** focuses on four distinct categories of causal factors.

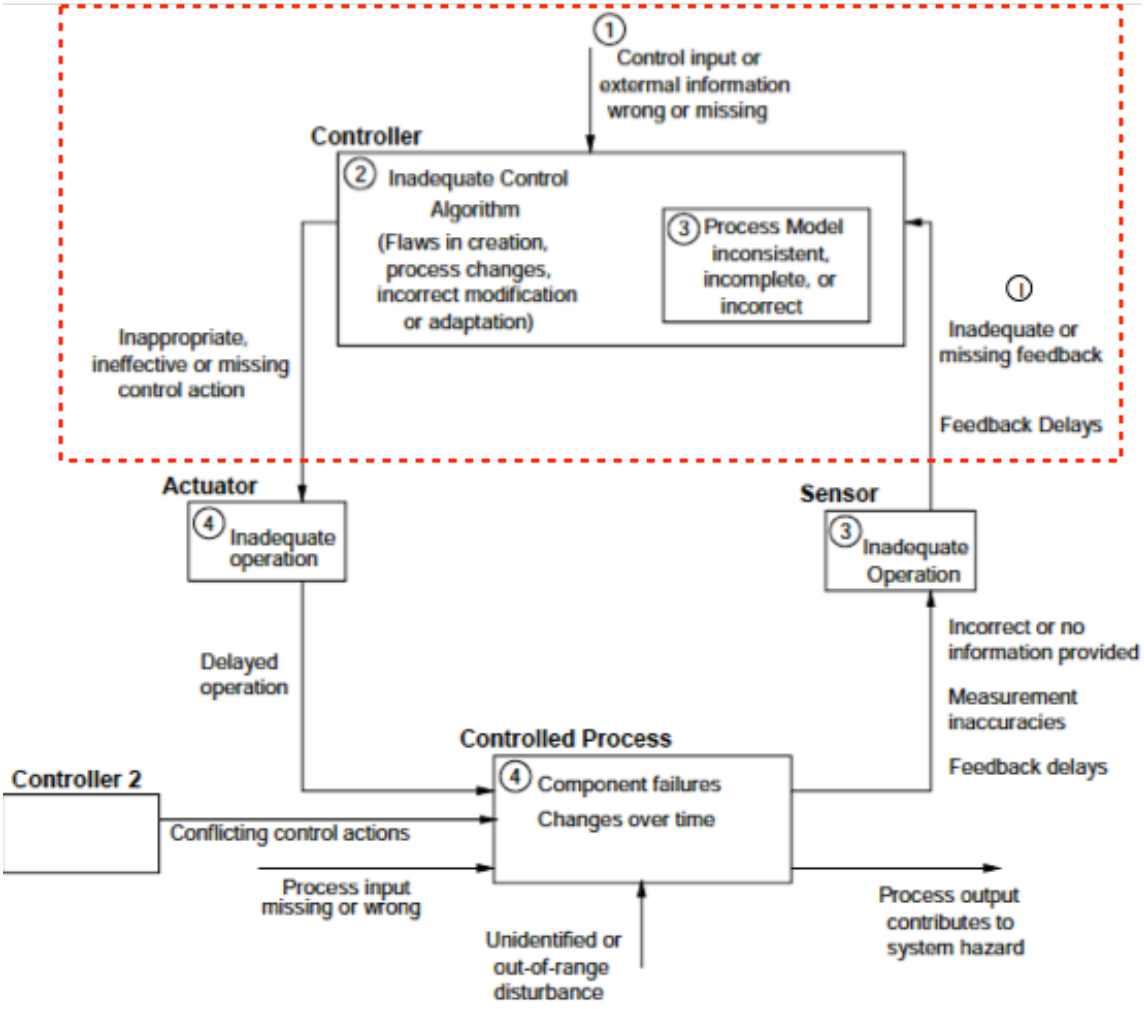

Figure 12. The Current Human-Controller Analysis [5]

As **Figure 12** shows, these four categories revolve around inadequate feedback, wrong control input or external information, inconsistent process models, and an inadequate control algorithm. Applying the updated human-controller model from the previous section, the causal-factor analysis now explicitly related to a *human controller* is seen in **Figure 13**.

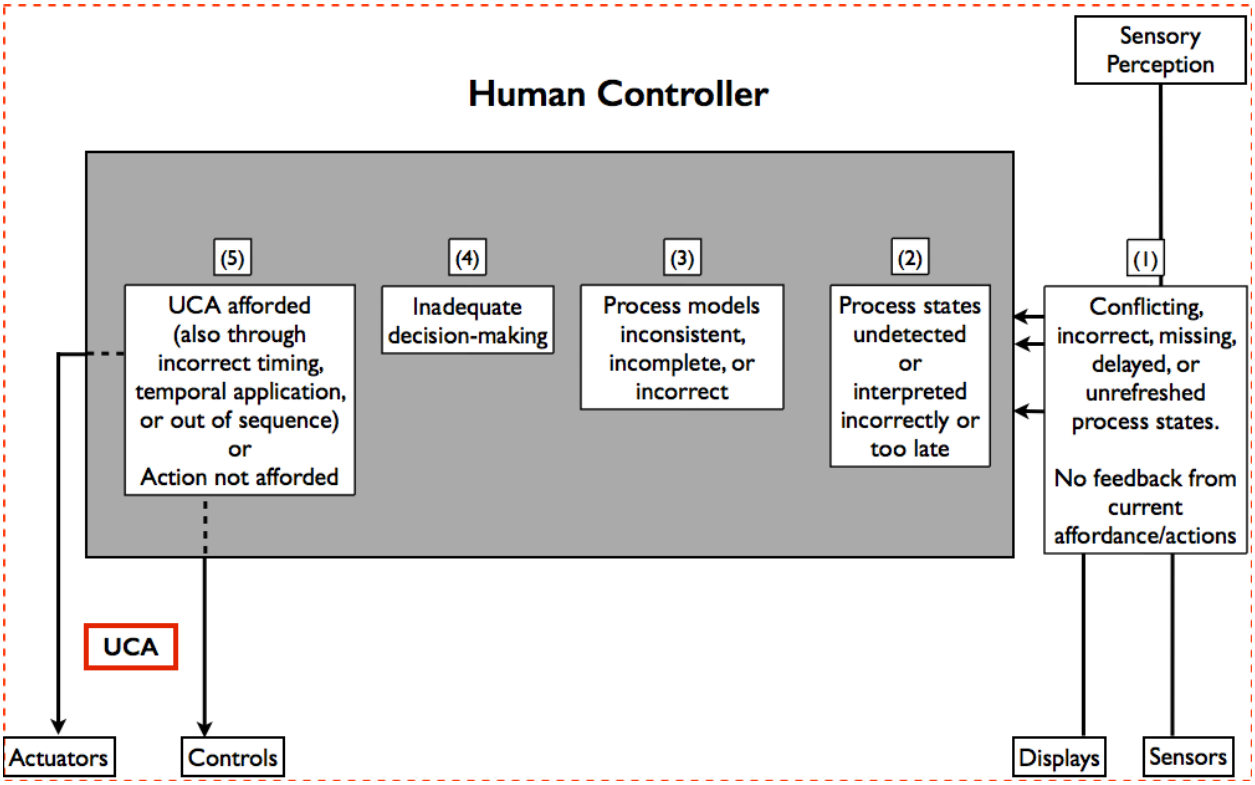

Figure 13. The Updated Human-Controller Analysis

As a whole, the updated human-controller analysis in **Figure 13** includes five distinct causal-factor categories that largely revolve around the process states for any given UCA. The five categories above directly stem from **Figure 11** through the distinct stages of "feedback," human cognition ("detection and interpretation," "mental models," and "decision-making"), and the "affordance" of action.

The first update in **Figure 13** resides in the analysis of flawed feedback as denoted by the (1), or first causal-factor category. This updated model combines all feedback from **Figure 11** (i.e., from the displays, sensors, and sensory perception) and *centers this causal-factor category of feedback on the process states themselves*. It is important to note that this first category analyzes the feedback in and of itself *before* it reaches the human controller and indicates whether or not that feedback, which is used to update the human controller about the process states for a given UCA, is conflicting, incorrect, missing, delayed, or unrefreshed. As Section 3.3 will detail, the analysis of this first category will methodically address each component of

feedback by focusing on each high- and lower-level process state. Furthermore, this first category also includes causal factors related to feedback from the affordance and action of a UCA despite there being no direct link between affordance, action, and feedback. After this feedback reaches the human, there are four more causal-factor categories that could lead to unsafe control.

The second category of causal factors seen in **Figure 13** focuses on the flawed detection and interpretation of feedback (2) *after* it reaches the human. In contrast to dissecting the feedback itself, this new stage explicitly analyzes potential flaws in the human detection and interpretation of feedback that could lead to unsafe control. In this manner, the distinction between feedback and the human detection and interpretation of that feedback is clarified to better organize the ensuing causal factors. Thus, in order to achieve system safety, the human controller must accurately detect and interpret the process states as they exist in reality. Anything other than the accurate detection and interpretation of the process states through feedback (displays, sensors, and the human's own sensory perception) will, when paired with a worst-case set of environmental conditions, lead to an accident.

While the process states are useful in determining *what* feedback to analyze, another reason for including the detection and interpretation category (2) is to address *how* the human receives the feedback. *What* feedback is flawed (1) is just as important as *how* the detection and interpretation of that feedback can become flawed (2). While the design of the system around the human controller is tightly coupled with system safety (e.g. the presentation of feedback, the layout of controls), this approach to analyzing *how* must be treated with caution because the goal of STPA is hazard analysis and not design. This new category will, however, accomplish hazard analysis by better informing the designers of how the feedback can become undetected or misinterpreted through the use of process states as the next section will elaborate.

The next two categories of inconsistent process models (3) and flawed decision-making (4) largely remain unchanged from the previous human-controller analysis in **Figure 12**. Aside from the updated semantics in (4), the *method* by which flawed process models (3) are generated has changed. In the previous human-controller analysis, there was no explicit means of

generating the causal factors in (3), but in this updated analysis, all causal factors in (3) will derive directly from the process states—detailed in Section 3.3.

Following the same logic as **Figure 11**, the last causal-factor category is that of inappropriate affordance (5). In this category, the causal factors that can lead to an accident stem from the inappropriate affordance or inaffordance of action that leads to a UCA. There is also the unseen link between affordance and feedback that might become severed and contribute to an unsafe control action. In either case, this updated human-controller analysis in now specifically includes a new category that magnifies these issues in affordance.

Altogether, the biggest changes in appearance in **Figure 13** were explicitly identifying the controller as human, grouping the feedback into one coherent category, and adding the "detection and interpretation" and "affordance" categories. More importantly, much of the subsequent causal-factor analysis will now be framed around the system and environmental properties that are necessary for system safety—the process states. Despite these improvements in appearance, the strongest and most distinctive update to this model is not in the new categories themselves, but in the method by which the causal factors are generated.

3.3 Application

3.3.1 Context Tables

Since the human-controller causal factors in **Figure 13** are grounded in the contextual process states, the application of this new Step 2 analysis first begins with the context tables from STPA Step 1 as outlined by Thomas in his systematic method [12, 13]. After the foundational context tables have been established with the relevant process states necessary for safe control, the next step will focus on choosing a UCA (unsafe control action) or group of UCA's for causal analysis and refining the process-state hierarchy.

3.3.2 Unsafe Control Actions and Process States

The purpose of the Step 2 causal analysis is to detail all causal factors and scenarios that could lead to an identified unsafe control action, so the next step begins by selecting the UCA's for a given control action. Considering a human controller that acts as a simple train door

controller for an example, **Table 5** details all of the UCA's that could occur with an "open door" control action given the simplified process states of "train motion," "emergency," and "train position" through the use of the systematic method.

Table 5. Context table for the "open door" control action [12]

Control Action	Train Motion	Emergency	Train Position	Hazardous control action?		
				If provided any time in this context	If provided too early in this context	If provided too late in this context
Door open command provided	Train is moving	No emergency	(doesn't matter)	**Yes**	**Yes**	**Yes**
Door open command provided	Train is moving	Emergency exists	(doesn't matter)	**Yes**	**Yes**	**Yes**
Door open command provided	Train is stopped	Emergency exists	(doesn't matter)	No	No	**Yes**
Door open command provided	Train is stopped	No emergency	Not aligned with platform	**Yes**	**Yes**	**Yes**
Door open command provided	Train is stopped	No emergency	Aligned with platform	No	No	No

In this example, the control action is providing the "open door" command and the UCA's are all of those contexts in **Table 5** for which the "open door" control action is hazardous. Once these UCA's are selected from the context tables, the focus then shifts to the associated process states. Tracing back to the Chapter 2 discussion regarding process-state hierarchy, this involves refining the high-level process states down to lower-level process states as necessary. Some top-level process states may not have a lower-level hierarchy, whereas other top-level process states may have multiple lower-levels of hierarchy. Note that the process states in **Table 5** are still left at the highest level of abstraction and "emergency" has not been refined down into a simple lower-level hierarchy of "smoke present," "fire present," and "toxic gas present". Furthering this example, it may be important in Step 2 causal-factor analysis to use the lower-level process states that compose "emergency," or perhaps this level of detail is unnecessary. What is important is that the top-level process states be broken down to whatever level of detail suits the

hazard analysis—a decision ultimately left to the safety engineer(s). The process-state hierarchy and feedback involved in this simple train door controller example are tabulated in **Table 6**.

Table 6. Process-State Hierarchy and Feedback

Process State	**1. Train Motion** (moving/stopped)	**2. Train Position** (aligned/not aligned)	**3. Emergency** (no/evacuation required)
Lower-level Process States			**3.1 Smoke present** (Y/N) **3.2 Fire present** (Y/N) **3.3 Toxic gas present** (Y/N)
Designed Feedback	**1. Train motion** - Speed sensor #1 - Speed sensor #2 - Speed sensor #3	**2. Train position** - Left platform sensor - Right platform sensor	**3. Emergency** **3.1 Smoke present** - Ionization smoke sensor - Optical smoke sensor **3.1 Fire present** - Engine compartment fire sensor - Passenger compartment fire sensor **3.1 Toxic gas present** - Toxic gas sensor

The values of each bolded process state in **Table 6** are shown in parentheses and in this case, each consist of only two values (e.g., train moving or train stopped). Below the process-state hierarchy are example sources of feedback that have been designed into the system to inform the human controller about these process states. This does not account for the raw sensory feedback the human receives, so for example, in addition to the three speed sensors that inform the human operator about the train's motion, the human also receives feedback from their own visual, proprioceptive, and vestibular senses. STPA Step 2 then begins after selecting the UCA(s) and the process-state hierarchies as illustrated in **Table 6**.

3.3.3 Human-Controller Causal-Factor Analysis (Step 2)

Referring back to the updated analysis in **Figure 13**, each of the five categories must be methodically analyzed for *each* unsafe control action. That is, each UCA in the context table must be analyzed independently unless there is enough overlap between a set of UCA's to warrant parallel UCA analysis, which Chapter 4 will demonstrate. Once a UCA has been selected for Step 2 analysis, the recommended place to start is with the flawed process models

(3). This category stems directly from the process states and will help to guide the other four categories. To note, the new approach to this category of flawed process models (3) is not unique to a human controller and in fact equally applies to an automated controller. "Flawed process models" mean that the controller (human or automated) thinks that a process state *is not* in a hazardous state when in fact, *it is*, and this could refer to any unsafe combination of the process states involved along with their subsequent lower-level hierarchies. This is to say that the process model variables (PMV's) in the controller's mind or software differ from the process states of reality. Take, for example, the unsafe control action of providing the "open door" command **when the train is not moving, there is no emergency, and the train is not aligned with the platform**, as seen in the second-to-last row of **Table 5**—assuming a human controller. The flawed process models (3) that could lead to this UCA are:

The human controller thinks that:
1. The train **is not** moving when **it is**, or
2. An emergency **exists** when one **does not exist**, or
3. The train **is aligned with the platform** when **it is not aligned with the platform**.

These three cases represent an "OR" combination but other UCA's may include an "AND" statement as well and is entirely system dependent. Also, the second process state of "emergency" could alternatively be written as:

The human controller thinks:
2.1. **Smoke is present** when **it is not**, or
2.2. **Fire is present** when **it is not**, or
2.3. **Toxic gas is present** when **it is not**.

In the case above, 2.1–2.3 leverage the process-state hierarchy of "emergency" to better detail the flawed mental models within the human's mind. After the flawed process models (3) are

detailed using the process-state hierarchy as necessary, the other four categories can be tackled in any order.

Since the process states are already in use, however, the next category will focus on the flaws in feedback (1), which also apply to both an automated and human controller. Recall that process states represent those real states in the environment or the system and that feedback is the channel through which those states reach the controller. Feedback used to inform the controller of any lower- or high-level process state may come from a solitary source or from multiple sources across multiple modes (i.e., visual, tactile, auditory, etc.), and flaws in this feedback can lead to an incorrect, missing, delayed, unrefreshed, or conflicted process state. In addition, there may be no link between human affordance, action, and feedback that could contribute to a UCA. The feedback can therefore be incorrect, missing, delayed, unrefreshed, or conflicted *before* it even reaches the controller. While the terms "incorrect," "missing," "delayed," or "unrefreshed" (not refreshed in the appropriate amount of time) are self-explanatory, the new definition of "*conflicted"* process states or feedback needs explanation and clarification. The value of a process state, regardless of where it lies in the hierarchy, is the combination of all feedback that is used to represent it. If that feedback comes from multiple sources, the feedback from one source can contradict the feedback from another source that in turn might lead to an overall *conflicted* process state value. Furthermore, lower-level process states may also conflict with each other through flaws in feedback, which can in turn lead to a *conflicted*, unknown, or undefined value of the high-level process state. Notice that these flaws in feedback (1) are all framed around the process states and *not* the feedback. This structure is part of the new approach in this updated Step 2 human-controller analysis that will help purposefully guide, organize, and detail the ensuing causal factors.

With regards to analyzing the flaws in feedback (1), this process involves: 1) listing the process states, their associated hierarchies, and sources of feedback, and then 2) methodically addressing each high- and lower-level process state with respect to the feedback channels that comprise it. In the train door controller example, the first high-level process state of "train motion" contains no lower-level process states and has four main sources of feedback: speed sensor #1, #2, #3, and the human's own sensory perception of motion. Focusing on this process

state of "train motion" exclusively, the underlying sources of causal factors that could lead to unsafe control are:

Train motion (**PS 1)**:
- is incorrect or missing.
- isn't refreshed in the appropriate amount of time.
- is conflicted.

Notice again that these causal factors are framed around the *process state* of "train motion" (abbreviated as **PS 1**) and not around the *feedback* that comprises "train motion." Given this structure, it becomes evident how the speed sensors could be incorrect, missing, or not refreshed in the appropriate amount of time, and perhaps more importantly how speed sensors #1–3 could conflict with each other or conflict with the human controller's own sensory perception of motion through changing visual fields or vestibular accelerations, for example. Using the same approach for the other high-level process states, the other two process states of "emergency" and "train position" can be analyzed similarly. The causal factors related to "emergency" could further be refined to look like:

Any lower-level **emergency (PS 2.1–2.3)**:
- is incorrect or missing.
- isn't refreshed in the appropriate amount of time.
- is conflicted which leads to an ambiguous **emergency (PS 2)**.

In addition, although not applicable to this example, the high-level process states can also conflict with each other and lead to unsafe control, meaning that a high-level **PS 1** could conflict with a high-level **PS 2**, etc., which could lead to a UCA (Chapter 4 will provide an example). It cannot be emphasized enough that the focus here is again with the process states and not the feedback—a shift that will help generate more specific safety constraints and aid designers' awareness of potential feedback flaws and conflicts. Perhaps in the previous example the

designers refine the speed sensor software or detail procedures for the human operator to follow in the event of speed sensor conflicts or disorientation—either way, the designers are aware of these issues and in position to improve system safety, the fundamental purpose of this analysis. As stated earlier, the emphasis on process states also makes the analysis of flawed feedback (1) and inconsistent process models (3) equally applicable to an automated controller, an unintended but welcome effect of this new approach to the human controller.

The steps in generating the causal factors in flawed detection and interpretation (2) will follow the same pattern as the previous category of flawed feedback (1), but are now exclusive to the human controller. The steps in this category are to: 1) define the process-state hierarchy, and 2) address each high- and lower-level process state with respect to their detection and interpretation by the human operator. Instead of dissecting the feedback itself, this category of flawed detection and interpretation (2) analyzes how correct feedback could not be detected or become interpreted incorrectly *after* it reaches the human. Revisiting the "emergency" high- and lower-level process states from the train door controller, the second step in this causal-factor category would be written as:

Any lower-level **emergency (PS 2.1 – 2.3)** OR their changes/updates:
- are not detected.
- are not interpreted correctly and leads to an inaccurate or conflicted understanding of the **emergency (PS 2)**.
- take too long to detect and interpret correctly.
- require too much attentional demand to detect and interpret correctly.

While the feedback representing all three of the lower-level "emergency" process states may all be correct, it matters none if the human cannot detect or interpret these states correctly (as the process states exist in reality) within the temporal and attentional constraints of safe operation. Interpretations of feedback that lead to conflicted process states are just as much a causal factor as if the feedback itself was incorrect or conflicted *before* it reaches the human, and this updated analysis clearly differentiates between the two.

Shifting away from the use of process states, the fourth category of inadequate decision-making (4) remains concentrated on the flawed mental processes through which the proposed UCA will be decided upon. Aside from the human-friendly change in title, the approach to this category (4) remains the same as it did in analyzing the "inadequate control algorithm" in the current analysis of **Figure 12**. While outside the scope of this thesis, there is much room for exploration in this black box of human cognition and decision-making that could aid in improving STPA hazard analysis in the future.

Finally, the last category of inappropriate affordance (5) is based on the UCA or group of UCA's being analyzed—instead of the process states —and the causal factors under this category come from the inappropriate affordance or inaffordance of action that leads to unsafe control. This can occur from either a human slip or a mistake in decision-making, but the resulting UCA is the same. Furthermore, the link between affordance, action, and feedback may not be present to let the human operator know they may be incorrectly affording a UCA. In the train door controller example, where the unsafe control action is providing the "open door" command **when the train is not moving, there is no emergency, and the train is not aligned with the platform**, the causal factor in affordance would be written as:

- The train conductor affords the opening of the train doors, through a slip or a mistake and isn't made aware of this through any mode of feedback.

When a causal factor such as this is turned around into a safety constraint and handed to the designers, many new design considerations will emerge, such as how to structure the control panel layout (with the opening of doors in mind), how to design the specific mechanism(s) for opening the doors, how to improve feedback loops from the opening of doors, or whether or not to design automation to prevent the train conductor from opening the doors if the train is **not moving**, **there is no emergency**, and **the train is not aligned with the platform**. These are all important design issues to consider, and this last category of (5) inadequate affordance helps reveal these causal factors to ultimately improve system safety.

As a whole, the updated human-controller analysis in **Figure 13** combined the analysis of feedback and created two new categories unique to the human controller in detection/interpretation and affordance. Moreover, the application of process states to this causal-factor analysis created a new structure in the categories of flawed feedback (1), flawed detection and interpretation (2), and process model inconsistencies (3), for which (1) and (3) were equally applicable to an automated controller. Chapter 4 will apply this new human-controller analysis to an example involving In-Trail Procedure, or ITP.

4 An Applied Example using In-Trail Procedure (ITP)

4.1 Overview

In 2008, a safety analysis report titled *DO-312 (Safety and Interoperability Requirements Document for the In-Trail Procedure in Oceanic Airspace (ATSA-ITP) Application)* was completed by the Radio Technical Commission for Aeronautics (RTCA) to provide "the minimum operational, safety, performance, and interoperability requirements" for ITP [4]. It was in response to this original safety analysis on ATSA-ITP (Airborne Traffic Situational Awareness In-Trail Procedure)—an analysis based on fault trees and probabilistic assessments—that a 2012 STPA report (*Safety Assurance in NextGen*) was conducted to compare traditional safety methods with one based on the systems-theoretic approach of STPA [20]. This chapter will detail the approaches to the human controller in both DO-312 and the 2012 STPA report before applying the updated methodology based on **Figure 13** to further extend the STPA analysis.

As part of the FAA's upcoming NextGen architecture, the primary objective of In-Trail Procedure is to enable aircraft to achieve flight level changes on a more frequent basis to improve flight efficiency. This new procedure reduces separation minima through enhanced aircraft position and velocity reporting by means of ADS-B—a GPS-enabled aircraft transponder—which thereby allows aircraft to perform climb-through or descend-through maneuvers in procedural airspace where no radar services are provided [21]. Onboard the flight deck is also a new set of avionics specifically intended to assist the flight crew in conducing In-Trail Procedure, referred to as the ITP Equipment. ITP itself is comprised of six different maneuvers [4]:

1. A following climb
2. A following descent
3. A leading climb
4. A leading descent
5. A combined leading-following climb

6. A combined leading-following descent

An example of an ITP following climb is seen in **Figure 14**.

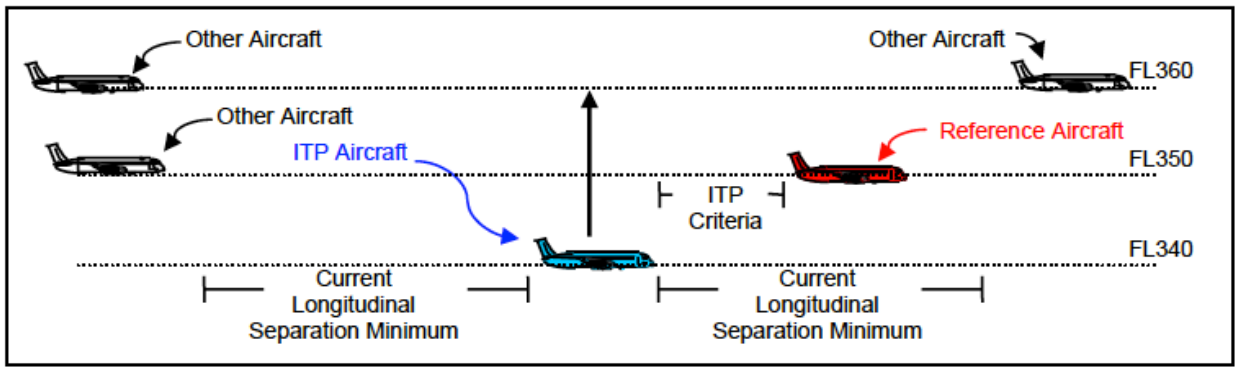

Figure 14. ITP Following Climb Maneuver [4]

In this example, both the ITP aircraft and the reference aircraft are equipped with ADS-B receive and transmit functions (ADS-B In and Out), so both aircraft as well as Air Traffic Control (ATC) are aware of the position and velocity data of the aircraft. With the known position and velocity of the reference aircraft through ADS-B, the flight crew of the ITP aircraft in this example is then responsible for four main items before the maneuver can be executed:

1. Checking and validating a set of ITP criteria for the following-climb maneuver with the assistance of the ITP Equipment.
2. Checking that the airspace is clear of conflicts with traffic, poor weather, turbulence, etc.
3. Requesting and receiving clearance from ATC to execute the following-climb maneuver.
4. Verifying the ITP criteria for the following climb maneuver with the assistance of the ITP Equipment.

After these four items have been met, it is then appropriate for the flight crew to execute the following-climb ITP maneuver to the requested flight level. Since this maneuver takes place in procedural airspace, the flight crew plays the largest role in ensuring the maneuver is executed in

a safe state—with ATC serving as a balance in verifying the ITP criteria and maneuver for the flight crew. It was due to the heavy involvement of the flight crew and hence the human controller in the execution of this ITP procedure that this example was ripe for human-controller analysis.

4.2 Human Error Analysis in DO-312

4.2.1 The DO-312 Safety Approach

DO-312 analyzes the entirety of ITP, including all human control elements, through a method called the Operational Safety Assessment (OSA). **Figure 15** provides an overview of the OSA and follows with a brief description from DO-312 [4].

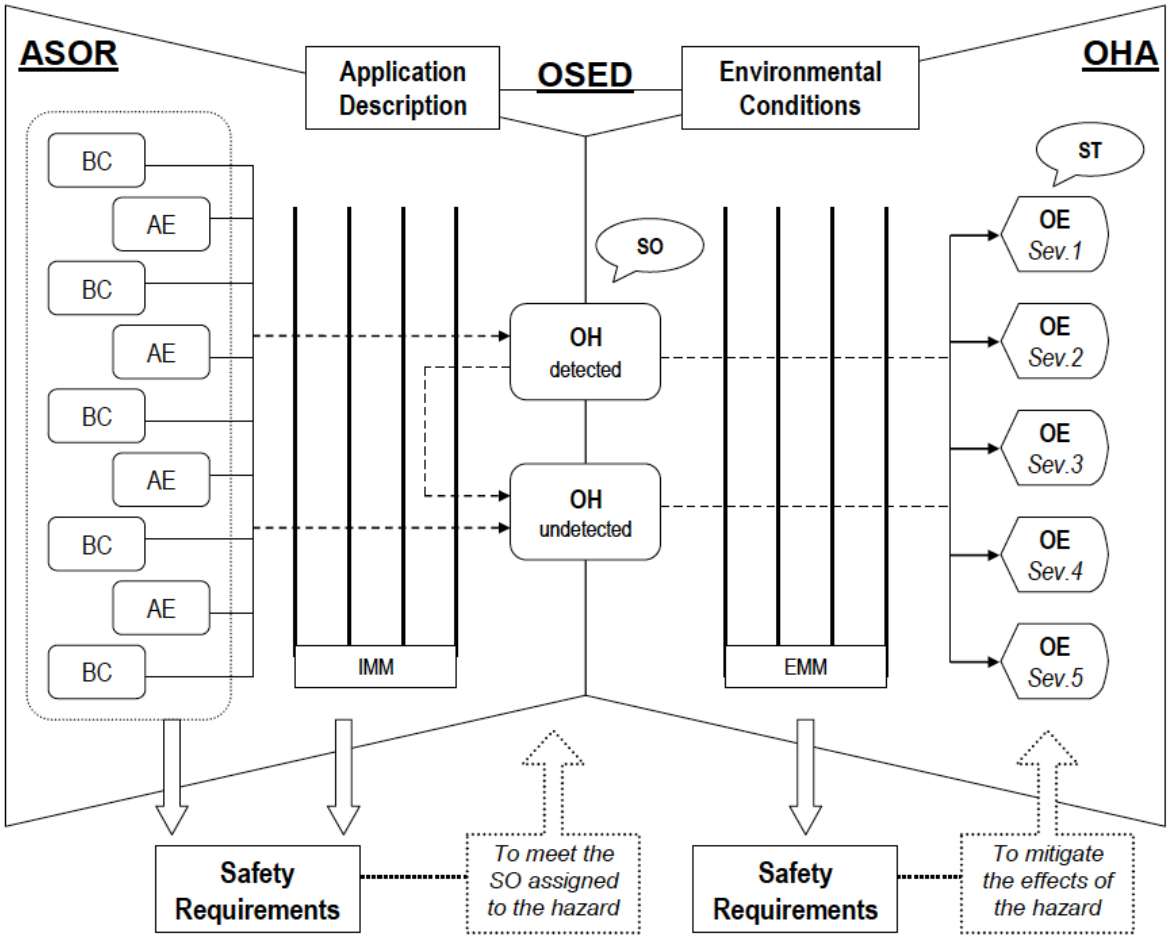

Figure 15. OSA Process Overview [4]

- In the center of the model stands the Operational Hazard (OH), both expressed for the detected and undetected case at the boundary of the application. Hazards are identified by operational personnel using the application description and associated phases and actions as a reference, along with a consideration of potential abnormal events.
- On the right-hand side resides the Operational Hazard Assessment (OHA), from the boundary of the application up to the operational effects on the airspace. The OHA objective is to set the Safety Objective for the OH (for both the detected and undetected case). External Mitigation Means, identified in the OHA and used in the determination of the Safety Objectives, are converted into Operational Requirements in the OSED.
- The left-hand side depicts the Allocation of Safety Objectives and Requirements

(ASOR) process, located inside the application. The objective of this activity is to allocate safety requirements to the airborne and ground domain in order to meet the safety objectives for each operational hazard. This is achieved by the identification of the Basic Causes leading to each hazard, their combination (shown in a Fault Tree) and the derived requirements. Internal Mitigation Means are identified to ensure the Safety Objectives are met—these become Safety Requirements if technical or Operational Requirements if procedural.

In contrast to usual definitions of system hazards that involve an accident or a general incident, the Operational Hazards (OH) in DO-312 are defined as those events "that may arise when the system is in a faulted mode [4]." The precise definition of a "faulted mode" is never clarified, however. The events that lead to an OH are called Basic Causes (BCs) and Abnormal Events (AEs) and can either be "system failures, *human errors*, procedures dysfunctions, or failures and conditions external to the application itself [4]." The hazard analysis technique within the Operational Safety Assessment is the Operational Hazard Analysis, or OHA, shown on the right in **Figure 15**. The purpose of the OHA is to essentially identify these events that could lead to an Operational Hazard and classify probabilities of their occurrence. The four steps to conducting an OHA are outlined in **Figure 16**.

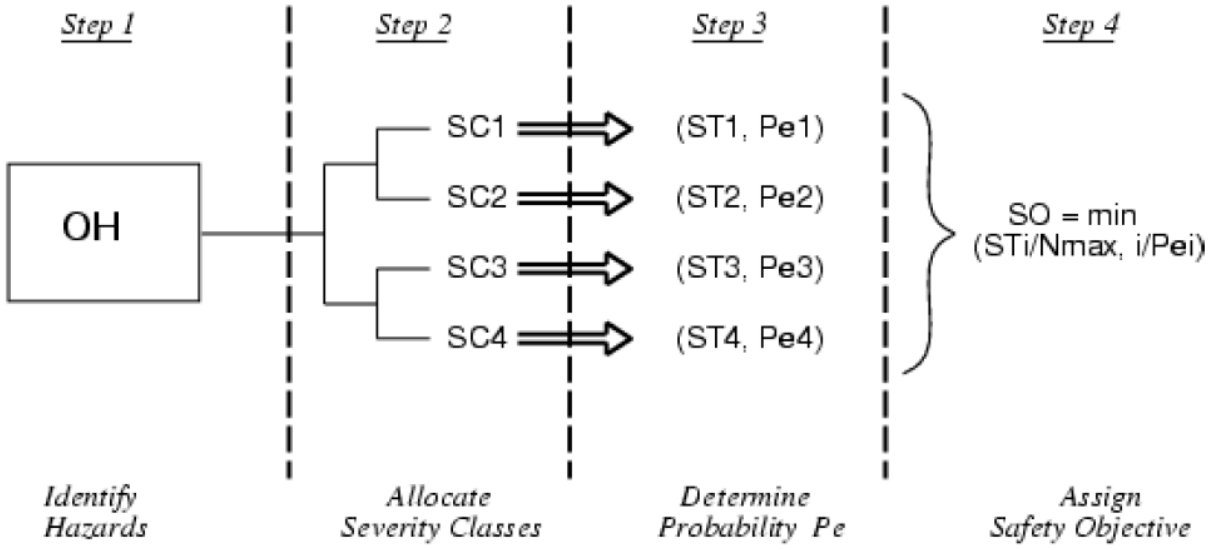

Figure 16. OHA Steps [4]

The first step in the OHA is to compile all the events that could lead to an Operational Hazard, whether the events are Basic Causes, Abnormal Events found through failure modes, or scenarios created by expert analysis—in this case involving operational controllers and pilots. The second step is to describe the operational environment and classify hazards based on formal fault tree logic. Following this, the third step is to determine and assign probabilities to these fault trees through Probabilities of Effect (Pe) and apportion the Air Traffic Management (ATM) risk budget (in this case) based on a Risk Classification Scheme (RCS). Finally, the last step in the OHA is to assign a safety objective that specifies "the maximum acceptable frequency of the occurrence of the hazard [4]." As evidenced by this linear event-based logic, the approach in the Operational Hazard Analysis is founded on the linear chain-of-events accident causality model, and with it, all the assumptions about the human controller.

4.2.2 Human Error in DO-312

Human errors are first identified in DO-312 by starting with an Operational Hazard and working top-down through a fault tree to identify the possible linear pathways to failure. Once a source of error or failure related to the human, ITP aircraft, reference aircraft, ground (Air Traffic Control), or environment (e.g., loss of GPS signal) is discovered, it is identified as a Basic Cause and the fault tree "branch" ends. Since OHA assumes reliability is synonymous with safety and that the reliability of the human can be calculable or at least estimated, "human error" is therefore framed through probability. The estimation of these probabilities in DO-312 arose from discussions involving pilots, controllers and operations experts and was classified according to **Table 7** using similar measures from the EUROCONTROL ATM standard [22].

Table 7. DO-312 Human Error Estimations [4]

Qualitative Frequency	Quantitative Probability
Very Often	1E-01 to 1E-02
Often	1E-02 to 1E-03
Rare	1E-03 to 1E-04
Very Rare	Less than 1E-04

The OHA itself methodically worked through six identified high-level Operational Hazards outlined in **Table 8** and refined them down into sub-hazards where appropriate. OH_2, for example, was refined into 11 distinct sub-hazards depending on which ITP criterion were met, whether the sub-hazard was detected or undetected by the flight crew, and whether or not the sub-hazards had a common cause or were independent.

Table 8. OH Descriptions [4]

OH Reference	OH Description
OH_1	Interruption of an ITP maneuver. Interruption that prevents successful completion of ITP. Flight crew abandons the maneuver. (Conditions external to the application such as an aircraft system failure or because of unintentional misuse of the ITP Equipment during an ITP maneuver requires the flight crew to abandon the maneuver and follow Regional Contingency Procedures.)
OH_2	Execution of an ITP clearance not compliant with ITP Criteria.
OH_3	ITP request not accepted by ATC. (flight crew requests ITP but the request is denied by ATC.)
OH_4	Rejection by the flight crew of an ITP clearance not compliant with the ITP Criteria.
OH_5	Rejection by the flight crew of an ITP clearance compliant with the ITP Criteria.
OH_6	Incorrect execution of an ITP maneuver. (Incorrect execution of an ITP maneuver by the flight crew by leveling off at the wrong Flight Level or Delaying the initiation of the ITP climb/descent.)

Interestingly enough, only four of these six high level OH's were analyzed in DO-312 since OH_4 and OH_5 were both determined in the report to have "no safety impact" on the ITP maneuver, which again raises the issue of how these OH definitions of a "faulted mode" specifically relate to safety [4]. Moving on to the human error analysis, an example of a sub-

hazard analysis of OH_2 is shown in **Figure 17** to highlight how human error is actually represented in DO-312.

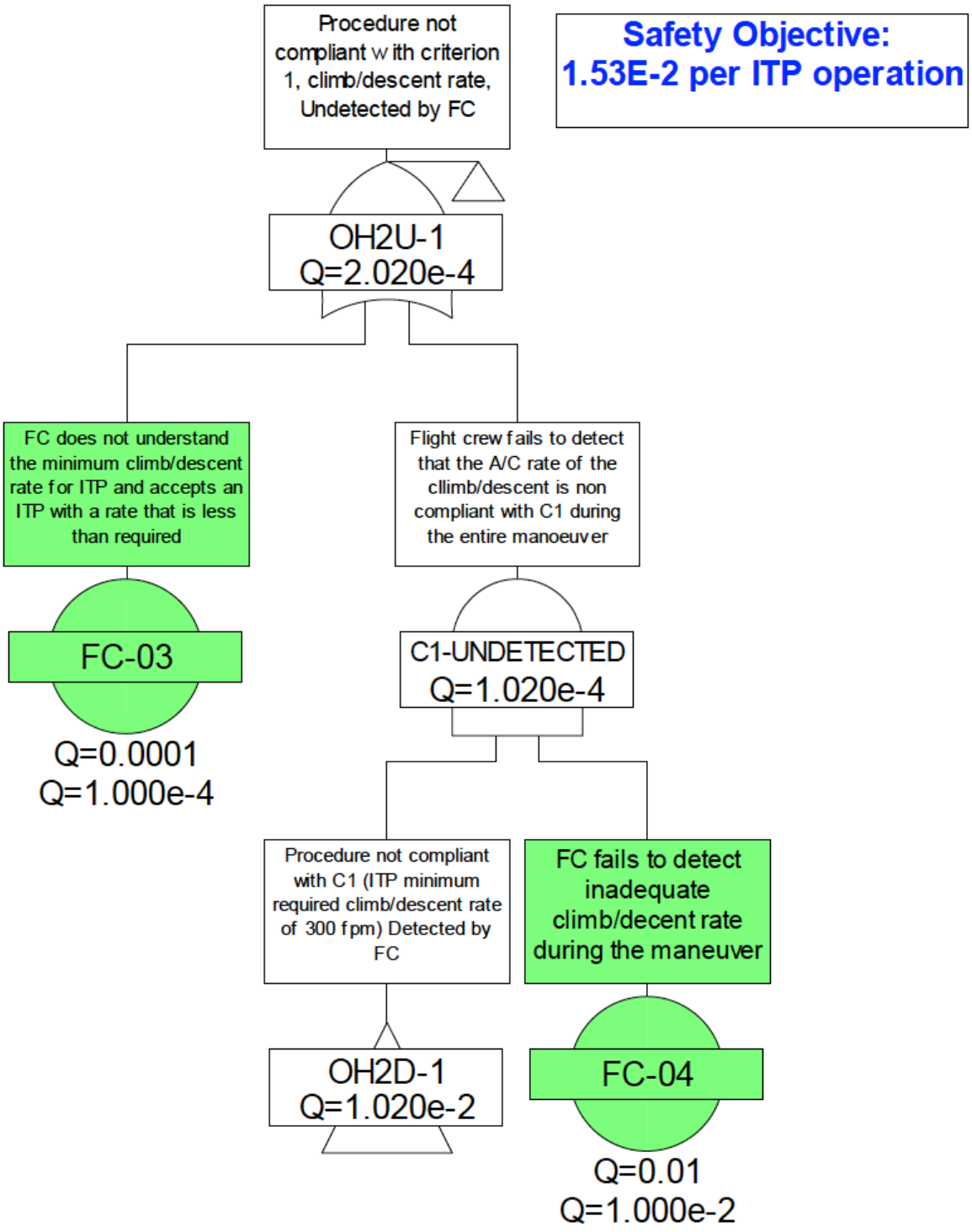

Figure 17. Execution of ITP with Non-Compliant Vertical Speed, Undetected by Flight Crew [4]

In this scenario in **Figure 17**, the OHA analyzes through fault tree logic how the ITP procedure, when not compliant with a vertical speed requirement (Criterion 1), can remain undetected by the flight crew (FC) during the ITP maneuver. Using the human error estimations from **Table 7**, the flight crew's misunderstanding of Criterion 1 (FC-03) was assumed to occur no more than Very Rare and their "failure" to detect inadequate vertical speed (FC-04) was assumed to occur no more than Often [4]. Thus, ITP execution without the required vertical speed can *only* occur when the flight crew randomly fails to detect improper vertical speed (FC-04) and when they randomly do not understand the vertical speed requirements (FC-03). There are many more examples similar to this in DO-312 and although these human error estimations were made in an earnest attempt to categorize the hazards associated with this scenario, they are fundamentally restricted in their analysis by their linear event-chain philosophy as discussed in Section 2.1.1.

4.3 Human-Controller Analysis in STPA

Shifting now to STPA, this section will focus on the human-controller analysis in the 2012 STPA report and then apply the updated human-controller analysis from **Figure 13** to further extend the results. Before diving into the STPA Step 2 analysis of the human controller, however, it is imperative to define the basic accidents, hazards, and safety control structure from the System Engineering Foundation as well describe the specific unsafe control actions (UCA's) that will be analyzed. As the 2012 STPA report detailed, the accidents under consideration are human death or injury and the high-level hazards are [20]:

H-1: A pair of controlled aircraft violate minimum separation standards.
H-2: Aircraft enters unsafe atmospheric region.
H-3: Aircraft enters uncontrolled state.
H-4: Aircraft enters unsafe attitude (excessive turbulence or pitch/roll/yaw that causes passenger injury but not necessarily aircraft loss).
H-5: Aircraft enters a prohibited area.

Hazards are again defined as “a system state or set of conditions, that, together with a particular set of worst-case environmental conditions, will lead to an accident [5].” The entire safety control structure for the ATSA-ITP system is outlined in **Figure 18**.

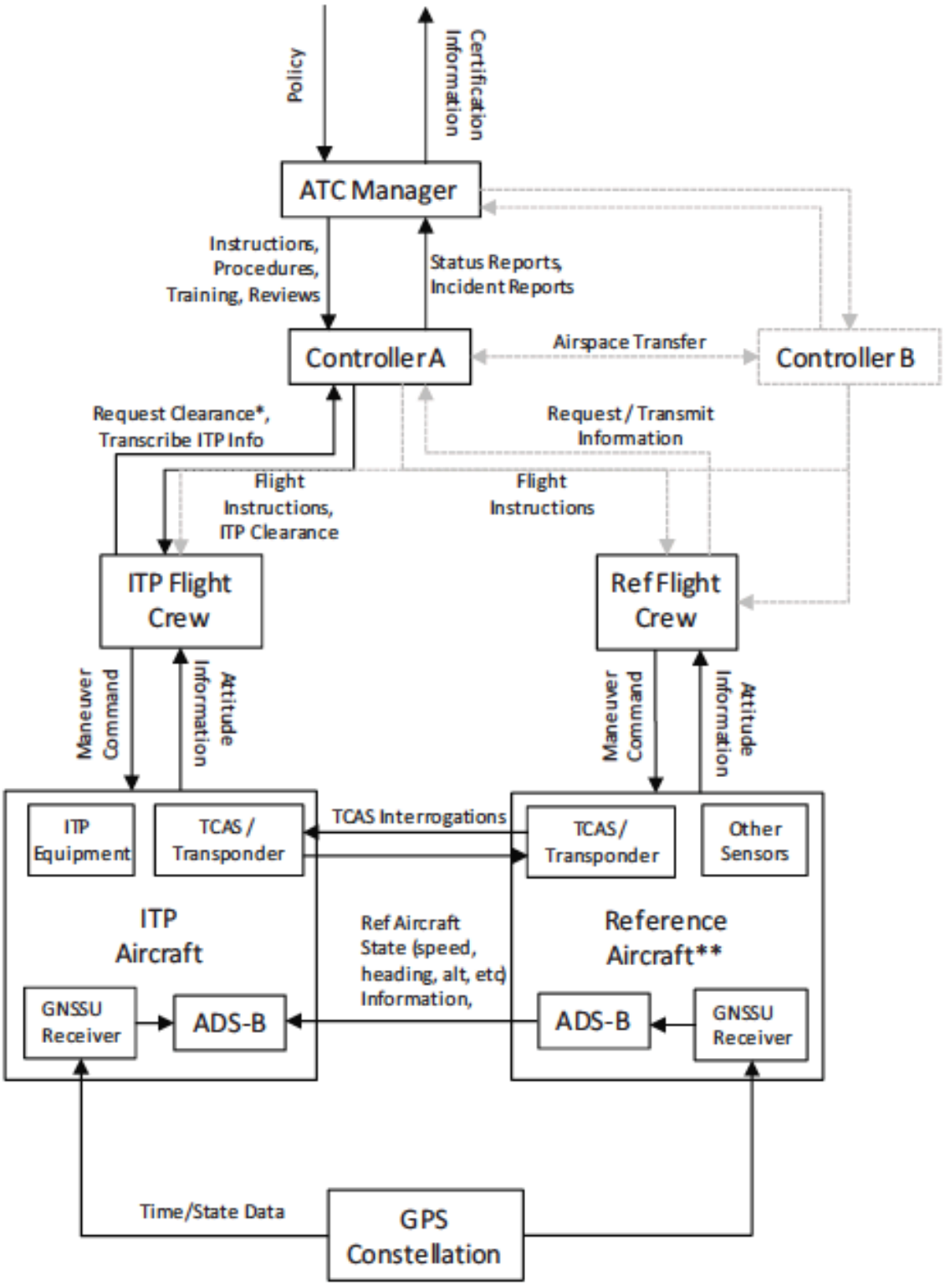

Figure 18. Safety Control Structure for ATSA-ITP [20]

From this foundational view of the accidents, hazards, and safety control structure, a set of UCA's specific to the human controller were selected.

4.3.1 Process States and Unsafe Control Actions

The UCA's chosen for this analysis all derived from the flight crew's execution of In-Trail Procedure. Due to a somewhat ambiguous definition, however, "execute ITP" was refined to include the stages of the "initiation" and "continuation of ITP," and omitted the "termination of ITP" in this example. Following the systematic method of STPA Step 1, the process states (**PS's**) for the initiation and continuation of ITP had to first be defined:

PS 1: ITP criteria
PS 2: ATC clearance
PS 3: Airspace model

The **first PS** of **ITP criteria** are actually "[a set of 10] conditions that must be satisfied prior to initiating or executing an ITP clearance," and include criterion such as the ITP distance between aircraft, vertical speed requirements, Mach differentials, data integrity, etc. [4]. As ITP is currently designed, these **ITP criteria** must be checked and validated by the flight crew before requesting ATC clearance and then validated again prior to executing the ITP maneuver. The **second PS** of **ATC clearance** is a system condition where ATC clearance is either approved or not. Since this example centers on the flight crew, the **ATC clearance** in this scenario is assumed to be correct—regardless of being approved or not. Finally, the **third high-level PS** of the **airspace model** is that state of the airspace outside of the immediate ITP maneuver. For ITP to be executed in a safe state, the ITP aircraft of **Figure 14** must not violate minimum separation standards with other aircraft as well as not fly into an unsafe atmospheric region and become uncontrollable—a direct relation to high-level hazards **H.1-H.4**. Given the 10 criteria within **PS 1** and the traffic/weather conditions within **PS 3**, both can be broken down into lower-level process states as shown in **Table 9**.

Table 9. The Process-State Hierarchy

High-level PS	**1. ITP Criteria** (met or not)	**2. ATC Clearance** (approved or not)	**3. Airspace Model** (clear or not)
Lower-level PS's	**1.1 Climb/Descent rate** (Y/N) **1.2 ITP distance*** (Y/N) **1.3 Ground speed differential*** (Y/N) **1.4 Mach differential** (Y/N) **1.5 Reference a/c maneuvering or expected to** (Y/N) **1.6 Vertical distance reqs** (Y/N) **1.7 Ownship data integrity** (Y/N) **1.8 Reference a/c data integrity** (Y/N) **1.9 Same track criteria*** (Y/N) **1.10 Requested flight level correct** (Y/N)	None	**3.1 Weather clear for ITP** (Y/N) **3.2 Clear of other traffic** (Y/N)

The first important detail in **Table 9** is that the high-level process states have two fundamental values: valid or invalid for ITP. It follows, then, that In-Trail Procedure is only valid when the **ITP criteria have been met, ATC clearance has been approved,** and **the airspace is clear**—and if any of these states are invalid ITP cannot be executed in a safe state. Secondly, each high-level process state can only be valid if every single lower-level process state within is valid (i.e., the **ITP criteria** (**PS 1**) can only be valid if **PS 1.1-1.10** are all valid), which the "(Y/N)" in **Table 9** is used to indicate. Lastly, the asterisks (*) next to **PS 1.2**, **1.3**, and **1.9** show which lower-level process states the ITP Equipment is planned to calculate on its own [4].

From this process-state hierarchy, the unsafe control actions selected for STPA Step 2 analysis were then generated and listed as shown in **Table 10**.

Table 10. The Selected UCA's for STPA Step 2

Control Action	UCA's with Process States
Initiate ITP	ITP initiated when **ITP criteria** (**PS 1**) have not been met ITP initiated when **ATC clearance** (**PS 2**) has not been approved ITP initiated when **airspace model** (**PS 3**) is not clear
Continue ITP	ITP continued with inappropriate **ITP criteria** (**PS 1**) ITP continued with revoked **ATC clearance** (**PS 2**) ITP continued with an **airspace model** (**PS 3**) that no longer permits ITP

All of the UCA's in **Table 10** were framed around the high-level process states and used for parallel STPA Step 2 analysis. That is, the human-controller causal factors from Step 2 were generated using all six of these UCA's at once and were not analyzed independently.

4.3.2 Human-Controller Analysis in the 2012 STPA Report

From the UCA's described in **Table 10**, the 2012 STPA report then generated the causal factors related to the human controller through the original template of **Figure 12**. All causal factors that were found to relate to the human controller are tabulated in **Table 11**.

Table 11. The Original Causal Factors Related to the Human Controller [20]

Hazard	**H.1-H.4**
UCA	- **ITP initiated** when any of **PS 1-3** are not met, approved, or clear for ITP - **ITP continued** when any of **PS 1-3** are no longer met, approved, or clear for ITP
Process-Model Link	**Causal Factors**
Inadequate control algorithm	Flight Crew: - does not correctly check that ITP is appropriate (that normal FL change could occur) - does not check that all ITP criteria are met - begins executing ITP prior to receiving approval - delays in executing ITP after receiving approval - does not re-verify that conditions have changed from when they were originally checked after receiving approval - does not confirm the established flight level after finishing the maneuver
Process model inconsistent	Flight Crew believes: - that their climb/descent capability is greater than it is - it has all ADS-B data for local traffic - ADS-B data to be accurate when it is not - ITP criteria (speed, distance, relative altitude, relative angle) to be different than it is - communication protocols with ATC to be different than they are - communication with nearby aircraft to be different than they are - in a different understanding of individual crew member responsibilities - weather/turbulence to be better than it is - ITP request to be approved when it is not - ATC approval to be recent when it is old
Inadequate Sensor Operation	- Flight crew does not understand or correctly apply the ITP data from the ITP equipment
Control input or external information wrong or missing	- Flight crew lacking information from ATC - ATC approval not on communication channel that FC is monitoring - ITP Equipment provides criteria data too late - ITP Equipment gives incorrect or ambiguous state information
Inadequate or missing feedback	- Change in the velocity/altitude/bearing of ownship not displayed to pilot - Change in the velocity/altitude/bearing of nearby ship not displayed to pilot - Proper aircraft identifier of nearby aircraft not displayed to pilot - FC does not receive communication from ATC - Flight crew does not receive local traffic information from ADS-B

As with any causal-factor generation in STPA, the Step 2 analysis must first connect to a specific set of UCA's (or single UCA) as well as the relevant high-level hazards, both of which are detailed in the first two rows of **Table 11**. The hazards all trace back to **H.1-H.4**, and the UCA's fundamentally translate to "the **initiation or continuation of ITP** when **any of the high-level**

process states are invalid for ITP." Located underneath this is the summary of all categories or "process model links" of the causal-factor analysis that trace back to **Figure 12**.

In this report, the causal factors in **Table 11** that were related to the flight crew stem from the categories of: inadequate control algorithms, process model inconsistencies, inadequate sensor operation, wrong external information or control inputs, and inadequate or missing feedback. To reiterate, the identified causal factors are not a comprehensive list of all those that were generated in the 2012 STPA report, but rather all those causal factors that specifically involved the flight crew for these UCA's. Notice that there is no relation to estimating human reliability or linear combinations of human failure that result in a hazardous state. The basic control loop template of **Figure 12** instead views the human controller (the flight crew) *within* the context of the system and controlled process occurring in the ecology to generate causal factors that could lead to accident scenarios. Furthermore, these causal factors lead to safety constraints (or requirements) that can be handed to system designers to ultimately design a safer system—one that does not operate in a hazardous state—and offer much more than random human error probabilities can convey. The methodology for using **Figure 12** in generating the causal factors in **Table 11**, however, is not clearly defined in STPA.

4.3.3 Extending the Human-Controller Analysis

In pursuit of extending this analysis, the new human-controller methodology was applied to the UCA's from **Table 10** to generate an entirely new set of causal factors. Following the process outlined in Section 3.3, this extended set of causal factors were derived according to the five categories from **Figure 13**. The causal-factor analysis first started with the inadequate process models (3), shown in **Table 12**.

Table 12. Inconsistent Process Model (3) Causal Factors

Hazard	**H.1-H.4**
UCA	- **ITP initiated** when any of **PS 1-3** are not met, approved, or clear for ITP - **ITP continued** when any of **PS 1-3** are no longer met, approved, or clear for ITP
Process-Model Link	**Extended Causal Factors**
(3) Process models inconsistent, incomplete, or incorrect	Flight Crew believes: - **ITP Criteria** (**PS 1**) has been met when it has not - **ATC clearance** (**PS 2**) to be valid when it is not - **airspace model** (**PS 3**) to be clear when it is not

In this category, any of the high-level process states are believed by the flight crew to not be in a hazardous state when in fact, they are. The biggest change in this category of flawed process models (3) stems from centering the causal factors on the abstract high-level process states, as this structure pinpoints exactly what beliefs on behalf of the flight crew may lead to an accident—an approach that would work equally as well for an automated controller. In addition, the three incorrect beliefs about the high-level process states are all part of a logical "OR" statement, so an accident can occur if the flight crew believes any one of these three things—an idea of logical simplification that may help future analyses.

The next category of inadequate decision-making (4) reflected no changes from the 2012 STPA report, aside from the change in title, so the next category of analysis shifted to the inappropriate affordance of ITP (5) as seen in **Table 13**.

Table 13. Inappropriate Affordance of ITP (5) Causal Factors

Hazard	**H.1-H.4**
UCA	- **ITP initiated** when any of **PS 1-3** are not met, approved, or clear for ITP - **ITP continued** when any of **PS 1-3** are no longer met, approved, or clear for ITP
Process-Model Link	**Extended Causal Factors**
(5) ITP inappropriately afforded	- Flight Crew inappropriately affords the initiation of ITP or continues to afford ITP, through a slip or mistake, and isn't made aware of this through feedback

This category of inappropriate affordance (5) in **Table 13** captured those ways by which the unsafe **initiation** and **continuation of ITP** could occur and to how the flight crew could not be made aware of this through feedback. This category also highlighted a missing link between affordance, action, and feedback—an unseen link that may bypass the controlled process and/or automation altogether and instead feed directly back to the human.

Following this, the causal-factor analysis then shifted to the treatment of flawed feedback (1) *before* it reaches the flight crew through the structure of the process states. As **Table 14** shows, each high-level process state and its lower-level hierarchy were methodically addressed.

Table 14. Flawed Feedback (1) Causal Factors

Hazard	**H.1-H.4**
UCA	- **ITP initiated** when any of **PS 1-3** are not met, approved, or clear for ITP - **ITP continued** when any of **PS 1-3** are no longer met, approved, or clear for ITP
Process-Model Link	**Extended Causal Factors**
(1) Conflicting, incorrect, missing, delayed, or unrefreshed process states. *No traceability to current affordance/actions*	Any of the **ITP criteria (PS 1.1-1.10)**: - are incorrect or missing - aren't refreshed in the appropriate amount of time - are in conflict which leads to an ambiguous **ITP criteria (PS 1)** **ATC clearance (PS 2)**: - is incorrect or missing - isn't provided in the appropriate amount of time - no longer remains valid (i.e. not refreshed in the appropriate amount of time) Either **Airspace model variable (PS 3.1 or 3.2)**: - is incorrect or missing - isn't refreshed in the appropriate amount of time - is in conflict which leads to an ambiguous **Airspace model (PS 3)** - There is a conflict between **ITP criteria**, **ATC approval**, and the **airspace model** (i.e. a conflict between **PS 1**, **PS 2**, and **PS 3**)

Using the **ITP criteria** (**PS 1.1-1.10**) from **Table 14** as an example, any of the feedback used to represent the lower-level process states (**PS 1.1-1.10**) can be, incorrect, missing, or unrefreshed in the appropriate amount of time. Not only this, but if any of the lower-level **ITP criteria** (**PS 1.1-1.10**) conflict with themselves through feedback (e.g., ITP distance readings from the Traffic

Collision Avoidance System (TCAS) and the ITP Equipment conflict with each other and lead to a conflicted **ITP distance** (**PS 1.2**)), the overall state of the high-level **ITP criteria** (**PS 1**) will be ambiguous, unknown, or undefined—i.e., conflicted. This same approach was then applied to the other process states of **ATC clearance** (**PS 2**) and the **airspace model** (**PS 3.1-3.2**). Another source of causal factors in flawed feedback (1) comes from conflicts between the high-level process states themselves which could lead to flawed mental models and decision-making (e.g., **ATC approves ITP clearance** even though the **ITP criteria are not met** and the **airspace model is not clear for ITP**. This could be further refined by asking: how could the flight crew initiate or continue ITP based on ATC's clearance **(PS 2)** when it conflicts with **PS 1** and **PS 3**?). A notable discovery in this process is that *conflicts* in feedback are now easily identifiable. Moreover, another important distinction in this approach is that *all* feedback to the flight crew is now categorized with respect to the process states, an approach that could also equally apply to an automated controller. In this manner, any number of ways by which the high- and lower-level process states become incorrect, missing, unrefreshed, or conflicted *through feedback* can be assessed—a large advantage for a system that has not even been fully designed yet.

Finally, the causal factors in the last remaining category of flawed detection and interpretation (2) were generated and are shown in **Table 15**.

Table 15. Flawed Detection and Interpretation (2) Causal Factors

Hazard	**H.1-H.4**
UCA	- **ITP initiated** when any of **PS 1-3** are not met, approved, or clear for ITP - **ITP continued** when any of **PS 1-3** are no longer met, approved, or clear for ITP
Process-Model Link	**Extended Causal Factors**
(2) Process states undetected or interpreted incorrectly or too late	Any of **ITP criteria** (**PS 1.1** - **PS 1.10**) OR their changes/updates: - are not detected - are not interpreted correctly and leads to inaccurate or conflicting understanding of the **ITP criteria (PS 1)** - take too long to detect and interpret correctly - require too much attentional demand to detect and interpret correctly **ATC clearance (PS 2)** or any change or update: - Anything but ATC clearance is detected and interpreted as a clearance - A revoke of ATC clearance is not detected and interpreted correctly Either **Airspace variable (PS 3.1 or 3.2)**: - is not detected - is not interpreted correctly and leads to inaccurate or conflicting understanding of the **Airspace (PS 3)** - takes too long to detect and interpret correctly - requires too much attentional demand to detect and interpret correctly - There is a conflict between the flight crew's interpretation of the **ITP criteria**, **ATC approval**, and the **airspace model** (i.e. a conflict between **PS 1**, **PS 2**, and **PS 3**)

The formation of these causal factors followed the same pattern as the last category of flawed feedback (1) by first starting with each high-level process state and its associated hierarchy. For example, any of the lower-level **ITP criteria** (**PS 1.1-1.10**) OR their changes and updates may not be detected and interpreted correctly in the appropriate amount of time, they may take too much attentional demand to do so, or they may be interpreted incorrectly which leads to an inaccurate or conflicted understanding of the high-level **ITP criteria** (**PS 1**) itself. An important understanding here is that while the feedback may be correct before it is detected by the flight crew, it matters none if the they cannot correctly interpret the feedback and understand the process states as they exist in reality. This misunderstanding of feedback is a causal factor that can easily lead to flaws in the process models (3) of the flight crew and an eventual unsafe

initiation or **continuation of ITP** when **any of the high-level process states are invalid**. Furthermore, just as the high-level process states could conflict with each other in flawed feedback (1), so too can the flight crew's interpretation of those high-level process states as **Table 15** lists.

4.3.4 Discussion of this Extension

From the original causal factors in the 2012 STPA report, the extension in this new approach changed the underlying structure and methods by which the causal factors were generated, which resulted in a longer and more structured list of causal factors and scenarios related to the flight crew. The backbone of this change was through the integration of the process states in the first three causal-factor categories of flawed feedback (1), flawed detection and interpretation (2) of feedback, and inconsistent process models (3). Much like the meaningful "state" variables in ecological situation are at the heart of what matters for effective human control, so too are the process states in STPA at the heart of what matters for safe control. Leveraging this principle, the entirety of feedback that reaches the human operator, to include their own sensory perception, is now analyzed in one specific category. An interesting finding is that the method for analyzing the categories of flawed feedback (1) and inconsistent process models (3) through the process states applies equally as well to an automated controller. Although the original intention was to improve the human-controller analysis, this is application to the automated controller is a welcome finding.

In addition to leveraging process states to frame parts of the causal-factor analysis, this new approach grounded in **Figure 13** revealed causal factors that had previously been difficult to identify—those related to *conflicts* in feedback. In the original methodology there were no clear means of establishing conflicts between feedback, or more importantly, *which conflicts mattered* (with respect to the process states). If the feedback or the flight crew's interpretation of it leads to a conflicted **model of the airspace**, or if the **ITP distance** is ambiguous, for example, the system can be designed to warn them or prescribe a course of action if this does occur. It is the entire purpose of STPA hazard analysis to hand the designers safety constraints that come from

detailed causal factors just like this, but in light of all five of these categorical analyses it is necessary to discuss the ramifications of this updated analysis.

4.3.5 Limitations of this Extension

Despite the improvements in clarity and organization in this updated analysis, there still remain some limitations to the new approach. The first criticism of this new analysis is that some of the generated causal factors can still be vague. For example, the category of affordance (5) highlighted a case where the flight crew inappropriately afforded the initiation or continuation of ITP and was not made aware of it through feedback. Aside from listing this as a causal factor, there is no further explanation to this statement. Although this is certainly an accurate criticism, it is important to remember that this analysis was conducted on a system *that has not even been designed yet*, so some of the causal factors are expected to be vague. For In-Trail Procedure, the automation has not been fully developed and the aircraft used in conducting ITP will vary—thus the affordance of ITP in each aircraft has still yet to be defined. With this analysis, however, the safety constraints that will be generated from these updated causal factors will help guide designers to construct a safe system in order that this causal factor is prevented.

Another criticism of this updated analysis is the poor link between affordance, action, and feedback. Aside from merely stating "no link between affordance, action, and feedback" as a type of causal factor, this new approach offers no more guidance. An assumption in STAMP theory is that all control actions go through the controlled process or automation before it eventually reconnects to the human controller through feedback from the sensors and displays. In most human-controlled systems this is not always the case, and the human operator may receive direct feedback from their affordance or action, such as when a pilot physically moves the yoke, for example. Perhaps there is another feedback arrow that comes directly the human controller's action, a possibility that should be explored in the future. In response to this criticism, the extended analysis at least identifies this as a causal factor now.

5 Conclusions

5.1 Extending the Human-Controller Methodology in STPA

Traditional safety approaches are grounded in linear logic and assume that component reliability equates to overall system safety. When these traditional hazard techniques analyze human controller components, if at all, they try in earnest to calculate or estimate the probability of human error independent of the designed system. The applicability of this approach, however, is fundamentally restricted when human behavior is characterized as a deviance from prescribed behavior, when it is viewed as a random event, and when it is removed from the context of the system and larger task environment.

In an entirely new approach to safety, the STAMP accident causality model applies systems and control theory to view safety as an *emergent property* of a system. STPA, the hazard analysis technique based on STAMP, captures nonlinear relationships between components to identify unsafe component interactions as well as the more traditional component failures. The analysis of the human controller within STPA looks far beyond "human error" and derives causal factors related to the human operator within the context of the system and their view of reality through the concept of process states and process model variables.

This goal of this thesis was to extend the causal-factor methodology of the human controller in STPA. This goal was approached by first exploring some of the literature related to ecological situation and cognitive modeling. By leveraging concepts from within these two areas, the STAMP model of the human controller was updated to clarify the feedback that reaches the human as well include the distinct stages of "detection and interpretation" and "affordance." This model was then applied to update the human-controller causal-factor analysis in STPA Step 2. In addition to analyzing the new causal-factor categories related to "detection and interpretation" and "affordance," all feedback is now analyzed in one distinct category before it reaches the controller. Most importantly, this new methodology adds structure through the use of process states that apply to *both* human and automated controllers.

These new methods of causal-factor generation were then applied to an example involving In-Trail Procedure. By extending a previous STPA analysis, this methodology added a

new categorical structure through the use of process states and generated new lists and definitions of causal factors that also made conflicts between feedback more easily identifiable. Overall, the research objective of extending the human-controller causal-factor analysis in STPA was accomplished through this new methodology as demonstrated in this example.

5.2 Future Work

This extension only presents an example of this new human-controller analysis and lacks the validation of a scientific comparison with the original causal-factor analysis in STPA. Before a comparison occurs, however, there are many avenues for improvement and future work. While this thesis applied basic cognitive models to the human controller, more investigation into cognitive abstraction levels (e.g., Rasmussen's SRK Framework or Abstraction Hierarchy) should be pursued to account for the high-level goals and decisions that affect lower-level operational actions. If investigated, there might also be additional ways to categorize feedback and control actions in STAMP through an abstraction hierarchy as well. This may ultimately enhance the overall perspective on human control and lead to new ways of analyzing the human controller in STPA with potential application to an automated controller. Furthermore, the formal specification language of SpecTRM-RL (Specification Tools and Requirements Methodology-Requirements Language) should be applied to this analysis when numerous process states and hierarchies are involved [23, 24] to simplify "AND/OR" combinations and increase the speed of future analyses.

References

[1] Dekker, S. *The Field Guide to Understanding Human Error*, Ashgate Publishing Limited, 2006.

[2] French Civil Aviation Safety Investigation Authority (BEA), *Final Report, Flight AF 447 Rio de Janeiro – Paris, 1st June 2009: English Edition*, 2012.

[3] The Dutch Safety Board, *Crashed During Approach, Boeing 737-800, Near Amsterdam Schiphol Airport*, 2009.

[4] RTCA, *Safety Performance and Interoperability Requirements Document for the In-Trail Procedure in the Oceanic Airspace (ATSA-ITP) Application*, DO-312, Washington DC, 2008.

[5] Leveson, N.G., *Engineering a Safer World: Systems Thinking Applied to Safety*, MIT Press, 2012.

[6] Heinrich, H.W., *Industrial Accident Prevention: A Scientific Approach*, McGraw-Hill Book Company, Inc., 1931.

[7] Reason, J., *The Contribution of Latent Human Failures to the Breakdown of Complex Systems*, Philosophical Transactions of the Royal Society of London. B, Biological Sciences 327, No. 1241, 1990.

[8] Swain, A.D., *Human Reliability Analysis: Need, Status, Trends and Limitations,* Reliability Engineering & System Safety, Vol 29, Issue 3: p. 301–313, 1990.

[9] Dekker, S., *Ten Questions About Human Error: A New View of Human Factors and System Safety*, Lawrence Erlbaum Associates, Inc., 2005.

[10] Dekker, S., *The Re-Invention of Human Error*, Human Factors and Aerospace Safety 1(3), 247-265, Ashgate Publishing, 2001.

[11] Flach, J., et al., *An Ecological Approach to Interface Design*, Proceedings of the Human Factors and Ergonomics Society 42nd Annual Meeting, 1998.

[12] *An STPA Primer, Version 1*, MIT SERL, 2014.
Available: http://sunnyday.mit.edu/STPA-Primer-v0.pdf

[13] Thomas, J., *Extending and Automating a Systems-Theoretic Hazard Analysis for Requirements Generation and Analysis*, Ph.D. Thesis, Engineering Systems Division, Massachusetts Institute of Technology, 2012.

[14] Flach, J.M. and F. Voorhorst, *What Matters*, Unpublished manuscript, Dayton, OH, 2012. Available: http://psych-scholar.wright.edu/flach/publications

[15] James, W., *Essays in Radical Empiricism, 1912 Edition*, Harvard University Press, 1976.

[16] Boyd, J., *The Essence of Winning and Losing*, John Boyd Compendium, Project on Government Oversight: Defense and the National Interest, 1995. Available: http://pogoarchives.org/m/dni/john_boyd_compendium/essence_of_winning_losing.pdf

[17] Boyd, J., *Patterns of Conflict*, John Boyd Compendium, Project on Government Oversight: Defense and the National Interest, 1986. Available: http://www.dnipogo.org/boyd/pdf/poc.pdf

[18] Rasmussen, J., *Skills, Rules, and Knowledge; Signals, Signs, and Symbols, and Other Distinctions in Human Performance Models*, IEEE Transactions on Systems, Man and Cybernetics, Vol. SMC-13, No. 3, 1983.

[19] Colvin, K., et al., *Is Pilots' Visual Scanning Adequate to Avoid Mid-Air Collisions?*, Proceedings of the 13th International Symposium on Aviation Psychology: p. 104-109, 2005.

[20] Fleming, C.H., et al., *Safety Assurance in NextGen*, NASA Technical Report NASA/CR-2012-217553, 2012.

[21] Federal Aviation Administration, *NextGen Implementation Plan*, 2013.

[22] EUROCONTROL. *Safety Regulatory Requirement, ESARR 4: Risk Assessment and Mitigation in ATM, Edition 1.0*, 2001.

[23] Leveson, N.G., et al., *Intent Specifications: An Approach to Building Human-Centered Specifications*, IEEE Trans. on Software Engineering, 2000.

[24] Leveson, N.G., *Completeness in Formal Specification Language Design for Process-Control Systems*, in Proceedings of the Third Workshop on Formal Methods in Software Practice, ACM: New York, NY: p. 75–87, 2000.