Technical Report

Modeling of Operator/Automation Authority

Ari Wilson[1]
NASA Langley Research Center
Research and Technology Directorate
Safety-Critical Avionics Systems Branch

Langley Aerospace Research Summer Scholars Program

[1]NASA Technical Mentor: Paul Miner

Abstract

Distributed fault-tolerant architectures for complex information processing and decision-making have been a recent focus of research in the quest for ultra-dependable safety-critical systems. SPIDER (Scalable Processor-Independent Design for Extended Reliability), developed at NASA Langley, is one such architecture. A large library of theories created for the analysis of this and other architectures already exists and has been verified by the formal verification system PVS and the model-checker SAL. We intend to extend this library to include the human-in-the-loop.

We have begun to attempt to formalize the link between human and automation, as well as between human and human using knowledge from human factors, distributed computing, and formal methods. This effort includes analysis of human behavior as a node in a distributed system. Our primary goal is to support dynamic function shifting between automation and human by defining an appropriate fault model for human interaction. This work includes a detailed study of pilot communication, general human capabilities with respect to automation, and modes of failure. A great emphasis was put on the ability to generalize our efforts to a wide variety of aircraft operations, automation capability, and operator roles. From these structures, overall system feasibility can be assessed and minimal, quantifiable fault principles have been derived and await review. After this is complete, work will begin on incorporating these structures into the language of the existing well-developed SPIDER theory.

Our work is preliminary and a safe and dependable distributed architecture based on our work might not be feasible. Even if it is feasible, it may not be seen in production aircraft or other safety-critical systems for many years. However, there is a growing need for investigation into transfer of duties between human and automation that we are now seeing in many fields. It is vital enough to our advance in safety-critical technologies that this research *should* continue to be pursued, in the interests of saving lives and providing for safer air transportation. Thus, NASA should continue to research this task in its pursuit of safe, reliable aviation technology.

Introduction

In the world of modern aviation, ever-improving technology in both avionics and aircraft design has led to a significant increase in aircraft safety as measured by a decrease in fatal accident rates[1]. A significant portion of this research into advanced technology has gone into creating more advanced and flexible computer systems in aircraft. A "glass cockpit" configuration, featuring adjustable and reconfigurable computer displays instead of mechanical instrumentation, is now common in many commercial and military aircraft. This technical advance has inspired much work in human-computer interaction related to keeping the pilot aware of relevant information to make appropriate and safe decisions. Pilots have come to rely on autopilots that have themselves grown more sophisticated and now can almost fly an entire flight by themselves, including takeoff and landing. Computer systems have also increased the safety of the mechanical structure of the plane. Fly-by-wire control systems have increased reliability and saved weight over older mechanical and hydraulic linkages. They also allow for precisely-measured tactile feedback to be given to the pilot.

However, the increased complexity of electronic systems within the airplane leads to many additional potential sources of failure. Indeed, many recent incidents have been caused by problems in either software or hardware, including the recent case of a Malaysian Air 777 where malfunctioning software nearly caused a major accident[2]. To alleviate this problem, independent and redundant electronics systems are built to serve as backups, but redundancy alone does not guarantee continued safe operation of the aircraft. After all, the first tire to go flat might just be the spare[3].

Computer science, however, has dealt with these types of problems for many years. Much work has been done on designing distributed fault-tolerant systems. Such a system can be defined so that tight error bounds on the number of undiagnosed errors in the system can be derived and proved, a guarantee that cannot be matched in scope via testing. Taking many ideas from this 30-year-old field of research, NASA researchers have created several fault-tolerant architectures that can handle small combinations of certain types of faults. One of the latest, called SPIDER (Scalable Processor-Independent Design for Extended Reliability), has been developed at NASA Langley over the last 5 years[TPMM05]. Using mechanical theorem-proving, many safety guarantees have been made and proved with respect to SPIDER. Reliability considerations are now a major concern for any airplane designer attempting to incorporate advanced electronic systems and architectures similar to SPIDER have and will find widespread use in the near future.

Even with the increasing use of these systems, safety cannot be guaranteed in many hazardous scenarios. The number one contributor to accidents and incidents found in the NTSB database over a recent four year period is pilot error of one sort or another[End95a] and the situation has not appreciably changed over the intervening years. Although there continue to be many efforts underway to decrease the pilot's workload and increase his or her situational awareness in the human factors domain [PB92] [PBB95] [EFJ+98], another approach has been gathering momentum in recent plans for the future of aviation safety. It is outlined in NASA's Intelligent Integrated Flight Deck Technologies (IIFDT) plan[YQ05] which calls for research into letting automation

[1] See `http://www.airdisaster.com/statistics/yearly.shtml`.

[2] Official report by the Australian Transportation Safety Board at `http://www.atsb.gov.au/publications/investigation_reports/2005/AAIR/aair200503722.aspx`.

[3] Quoting Roger Kieckhafer's Laws of Dependability, located at `http://www.ece.mtu.edu/faculty/rmkieckh/top-ten.htm`.

take on some of what has traditionally been the pilot's burden: essentially, decreasing the pilot's responsibility for decision-making in situations where it is demonstrable that he or she cannot or will not make an informed decision. The technical term for this is "dynamic function-shifting between human and automation." This idea of a man-computer symbiotic relationship and joint decision-making was prefigured in Licklider's seminal paper[Lic60].

We believe it is appropriate to approach this goal from the computer science theoretical perspective of fault-tolerant distributed systems, due to the large body of work in this area. In this way, we can define a human as a node within the larger system of the airplane. If our model is accurate, we can then define various safety and liveness properties about them that can be formally verified for a given architecture. Although, because of the still unknown complexities of human beings, this goal may ultimately prove to be unattainable. However, this level of quantification is not without precedent in both distributed systems and human factors literature. Using these two distinct bodies of research as a jumping-off point, as well as the mathematical formalisms embodied in SPIDER, we are attempting to provide specific models of human misbehavior, human communication, and have begun to give mathematical formalisms to both. From these, we are extracting preliminary safety and liveness properties that we think are minimal and sufficiently powerful to describe human safety behavior in the cockpit.

Justification for research

Several recent incidents illustrate the need for research into dynamic function-shifting between pilots and automation. In 1999, golfer Payne Stewart boarded a Learjet from Orlando to Texas. Along the way, cabin pressure decreased and hypoxia set in among both pilots and passengers. Eventually, all six people aboard died, and, after having been observed by Air National Guard F-16s, the plane crash-landed in an open field. The aircraft involved in this incident did not have much in the way of modern automation. However, it may be possible to design an aircraft so that if the pilot's unconscious state is observed and noted by automation, the aircraft lowers its altitude below 10,000 feet, below which cabin pressurization is not required. In fact, it may even land the plane. Had such technology been available in this incident, the lives of the passengers and crew might have been spared.

In 2005, another incident occurred involving Helios Airways Flight 522, a Boeing 737-31S flying from Cyprus to Athens. The flight experienced a loss of cabin air pressure as it entered Greek airspace. The copilot was observed by F-16s from the Hellenic Air Force to be slumped over the control console and the pilot was not visible. Many passengers, however, were still conscious, as oxygen masks had deployed in the cabin. Two people were observed to enter the cockpit and attempt to regain control of the aircraft, but they were unsuccessful. The aircraft engines quickly ran out of fuel and the jet crashed into a hill with the loss of all aboard, 121 in all.

One other accident stands out as an example of this type of scenario. A preliminary report from the NTSB[4] indicates that the pilot of a Predator unmanned aerial vehicle (UAV) was having problems with his command station and decided to switch to the other, equivalent station. He activated the autopilot of the aircraft and proceeded to switch stations with the other mission controller. Unfortunately, the fuel cutoff on the Predator had been activated and the autopilot did not have the authority to deactivate it. By the time the pilot reactivated his terminal, the drone was headed for

[4] Available at `http://www.ntsb.gov/NTSB/brief.asp?ev_id=20060509X00531&key=1`.

the ground, and by the time he realized what was happening, there was no way to stop it.

In these cases, the automation present on board these aircraft was not designed to or given authorization to override the pilot's commands and plot a safe course of action. As flight deck regulations and flight system architectures are currently designed, the pilot is the final arbiter in the cockpit. His decisions can only be advised against by the automation present, not overridden. Many lives can be saved if automation takes over in situations when pilots are incapacitated and unable to control their aircraft.

This research has implications beyond the field of aviation. Advances in automotive automation, including the recent development of collision warning and avoidance systems by Volvo and Mercedes, have increased the responsibility of automation from warning system to decision maker. The automation will override the driver's control on the gas pedal or steering wheel in case of an impending crash or unsafe lane change. This override could prove to be deadly in case of an erroneous detection on a high-speed highway. In a 2005 demonstration, Mercedes-Benz's crash-avoidance system failed in a public test in an environment not designed for by system designers. This fact was covered up by the automaker and the test was staged[5]. Further research is clearly needed to ensure such systems override human judgment only at such times as conditions are consistent with accurate, functioning sensors (as is done with a fault-tolerant architecture) or human impairment (this current work). In addition, some method is needed to tell the operator whether or not such a system is in operation. If the system is not engaged, then this fact needs to be made clear to the operator lest he or she take it for granted. At this time, neither of these concerns seems to have been investigated fully by the automakers or researchers developing such systems [JJG02].

Report organization

This report is broken down into several sections based on the general theme of the information contained therein. The background section gives information on distributed systems, fault-tolerance, the hybrid fault model, general human factors, and formal methods. The related work section details the nature of PVS, SPIDER, as well as Mica Endsley's work on situational awareness and Alan Pope's work on hazardous states of awareness. The analysis section gives our perspective on the nature of human communications, a summary of our human fault model, and a qualitative exposition of the effective differences between humans and automation. The results section gives our hypothesized human safety principles and justification as well as some preliminary safety and liveness properties. The future work and conclusion sections explain what this research has taught us, what remains to be done, and where we think it is headed.

Background

The following few sections give the necessary background to understand the context of our work. It can be skipped if one has a general understanding of the theory of distributed systems, formal methods, and human factors.

[5] See `http://www.drive.com.au/editorial/ArticleDetail.aspx?ArticleID=10734`.

Distributed systems and fault-tolerance

Distributed systems are heterogeneous computer systems with some measure of independent processing capability connected by a link or links that has or have neither infinite bandwidth nor zero latency [ALRL04]. They are prevalent in modern aircraft: aircraft have at least one, perhaps several, flight computers and many auxiliary digital systems including sensors, autopilot, etc. These together form a distributed system. Types of communication links may vary, however. A *bus* is a broadcast medium in which one message is received by many *nodes*, which are defined to be any of the elements of a distributed system. This may occur when a sensor wants to send its information to many flight computers. It is a basic assumption made of the nature of communication necessary for the operation of NASA's SPIDER architecture. In other systems, there can also be *point-to-point* communication, where one node sends a message to exactly one other node.

Even in the presence of extensive financial support and design time design is never perfect [AAS98]. Therefore, with a heterogeneous mix of complex components there is a non-vanishing chance that some component will *fault* in some manner. In the context of distributed systems, this fault can be characterized by the outputs it sends to other nodes in the system. If the outputs of the system as a whole fail to fall within an established bound, the fault will become a *failure* and the overall system state is in *error*. If the system in question was a distributed Internet server, perhaps the system could be restarted with an acceptable loss of service. However, in safety-critical domains like aircraft systems, no loss of service is ever acceptable, as it may lead to deaths or injuries for human beings.

Therefore, some method of diagnosing and tolerating faults must be developed – *fault tolerance*. This can be and is implemented in two main modes - *error detection* and *system recovery*. Another step that is often taken in creating an architecture is dividing system components into various failure or error *containment regions*, guaranteeing that failures or errors will not propagate beyond the limits of the zone in question. In addition, if a sufficient amount of *redundancy*, or duplication of system resources, is available, some undiagnosed errors can be tolerated [ALRL04].

In the creation of an actual architecture attempting to implement fault tolerance, protocols and algorithms must be designed for the common usage of distributed systems, including such areas as: *system initialization*, *scheduling*, *clock synchronization*, *diagnosis*, *isolation of malfunctioning components*, *reconfiguration*, *reinitialization*, and *interactive consistency*. Current architectures, including SPIDER, provide special means to deal with recovery from short-lived, or *transient*, faults. As one can imagine, it is very difficult to provide hard guarantees on system safety in all of these areas. In many of them, active research is still being pursued.

Hybrid fault model

As research progressed in distributed systems, it became evident that there are several classes of faults distinguished by the level of redundancy a system needs to tolerate such a fault. The first categorization lumped all faults into one category: Byzantine[LSP82]. *Byzantine faults* are those that can potentially send conflicting information to different parts of the system. Lamport et al. proved that, given n Byzantine-faulting nodes, $3n + 1$ nodes are necessary and sufficient to guarantee interactive consistency (*agreement* and *validity*) of results.

One established fault model is Omissive/Transmissive Hybrid-5 (OTH-5), which categorizes faults into five categories: transmissive asymmetric, strictly omissive asymmetric, omissive sym-

metric, transmissive symmetric, and benign [AK00]. A *transmissive asymmetric* fault can exhibit any form of arbitrary asymmetric behavior and is equivalent to a Byzantine fault. A *strictly omissive asymmetric* fault sends a single correct value to some processes and no value to all other processes, but cannot transmit an erroneous value to any receiver. An *omissive symmetric* fault fails to deliver any value to any receiving node. A *transmissive symmetric* fault delivers a single erroneous value to all processes. A *benign* fault is one that has been previously diagnosed by all nodes as faulty. A faulty node that does not send any messages to any receiver may be referred to as a *fail silent* node. The proposed OTH-6 model by Weber [Web06] decomposes the transmissive asymmetric fault mode into two distinct modes. A *single error omissive asymmetric* fault is one where some nodes receive the same erroneous value while other nodes receive no value. All other types of transmissive asymmetric faults are now referred to as *fully transmissive asymmetric* faults.

As some fault modes may be unlikely in a given application, fewer nodes may be needed to tolerate a desired number of faults. In our work, the hybrid fault model is used to evaluate our human failure hierarchy in terms useful for the analysis of distributed systems.

Human factors

Human factors is an umbrella term encompassing the interaction of human beings with products, tools, procedures and processes. Current research involves human performance, technology design, and human-computer interaction. A large body of work exists on human factors in aviation, mostly dealing with modeling pilot responsibilities and factors influencing command decisions in the cockpit. Some of this work is discussed in the related work section of this paper and was a large influence on our analysis of human communication and failures in the cockpit.

Formal methods

Formal methods is the use of mathematical techniques to specify, develop, and verify hardware or software systems. It is essential for safety-critical systems, as it can reveal flaws in complicated machinery that extensive testing may not find. As understood in the safety-critical domain, this mainly deals with developing formal specifications and verifying properties that should hold in order for the system to maintain safe operation. This is generally done in one of two ways: either using *model checking*, where the abstracted finite state space of a system is explored, or *theorem proving*, where logical inferences are made using the developer's understanding of the system. Both methods have their appropriate areas of application. The NASA SPIDER architecture makes heavy use of formal methods to verify a specification of the system against an established body of formalized principles. Generally, the ideas of formal methods were considered throughout our design process, and were specifically used to develop our principles and properties of human functions in the cockpit in an appropriately rigorous way.

In general, the properties we attempt to define have been traditionally categorized in terms of either safety or liveness[NC00]. In an intuitive sense, a *safety property* is a property that states that in a given system, some type of a bad event shall not ever occur. A *liveness property* is a property that, if satisfied, guarantees that, eventually, a good event will happen in any execution of a system. It is sometimes not a simple task to determine intuitively whether a given property is either safety or liveness, so a formal definition has been given. A formal definition of safety and liveness involves sequences, sets, and functions involving specified events. Formal proofs of safety or liveness can be

difficult and unintuitive but are, in general, always possible. In our work, this distinction becomes important when the computer science-theoretical definition conflicts with the aerospace definition of safety property, which generally involves ensuring the safety of the passengers and crew aboard the aircraft. We shall attempt to avoid conflating the terms by specifying which definition we are referencing. Making the distinction between these two properties also becomes important when implementing them within the context of a formal verification system.

Related Work

Other projects or research work that inform our approach are described in the following three sections. Although to our knowledge there is no major work that attempts to tackle dynamic function-shifting between humans and automation with our approach, the following work does speak to dealing with testing, verifying, and understanding the dangers inherent in designing for safety-critical aviation systems.

SPIDER and PVS

SPIDER is a family of fault-tolerant, reconfigurable architectures that provide powerful mechanisms for integrating inter-dependent applications of differing criticalities[6]. The applications communicate through a time division multiple access (TDMA) bus known as the Reliable Optical Bus (ROBUS). ROBUS provides the basic functionality that is built upon for higher-level services and is designed to achieve a message throughput close to the available bandwidth[TPMM05]. It incorporates many protocols and algorithms to deal with the practical necessities for distributed architectures mentioned in that section, as well as other, more recent considerations.

PVS is a system that provides mechanical support for specification and verification, based on typed, higher-order logic. PVS consists of a specification language and associated theorem prover, and is accessed using Emacs[7]. NASA is one of the largest users of PVS and has used it to verify many algorithms and designs, including most recently SPIDER.

The extensive fault-tolerance library developed in PVS for SPIDER has formed the basis for our work in dynamic human/automation function allocation. It has informed our ideas on how much rigor and generalizability is needed in our work, as well as illustrating traditional fault-tolerance and architecture ideas that are applicable in the human domain. As well, our work may eventually result in extensions of the library, if such an objective proves achievable.

Situational awareness

Situational awareness (SA) is a concept that was first defined in the field of aviation but has become much more general in the human factors domain. There are many definitions in the literature, but the one accepted for use in this project is, as defined by Endsley: "the perception of elements in the environment within a volume of time and space, the comprehension of their meaning, and the projection of their status in the near future." Thus, the analyses created for this project define and use the following three levels of situational awareness:

[6]Taken from `http://shemesh.larc.nasa.gov/fm/spider/`.

[7]The materials for a full course on PVS is available from the NIA at: `http://research.nianet.org/~munoz/`.

1. Perception of elements in the environment.
2. Comprehension of the current situation.
3. Projection of future status.

Methods to quantitatively measure SA have been developed in various domain areas[End95b], including aviation[EFJ+98]. Research in SA is broad and active. Therefore, SA was chosen to help categorize our human failure model as such minimal, quantifiable guiding principles were needed.

Hazardous states of awareness

As considered by Pope [PB92] [PBB95], among others, there are three primary factors that contribute to conditions which may lead to pilot error: preoccupation, vigilance, and excessive absorption. *Preoccupation* is characterized by thought unrelated to matters in the current situation, while *vigilance* relates to a persons overall attention within a situation, and *excessive absorption* is defined by the exclusion of all but a few elements in the current environment. As with situational awareness, these three categories can aid in finely distinguishing hazard scenarios to the level required for formal analysis.

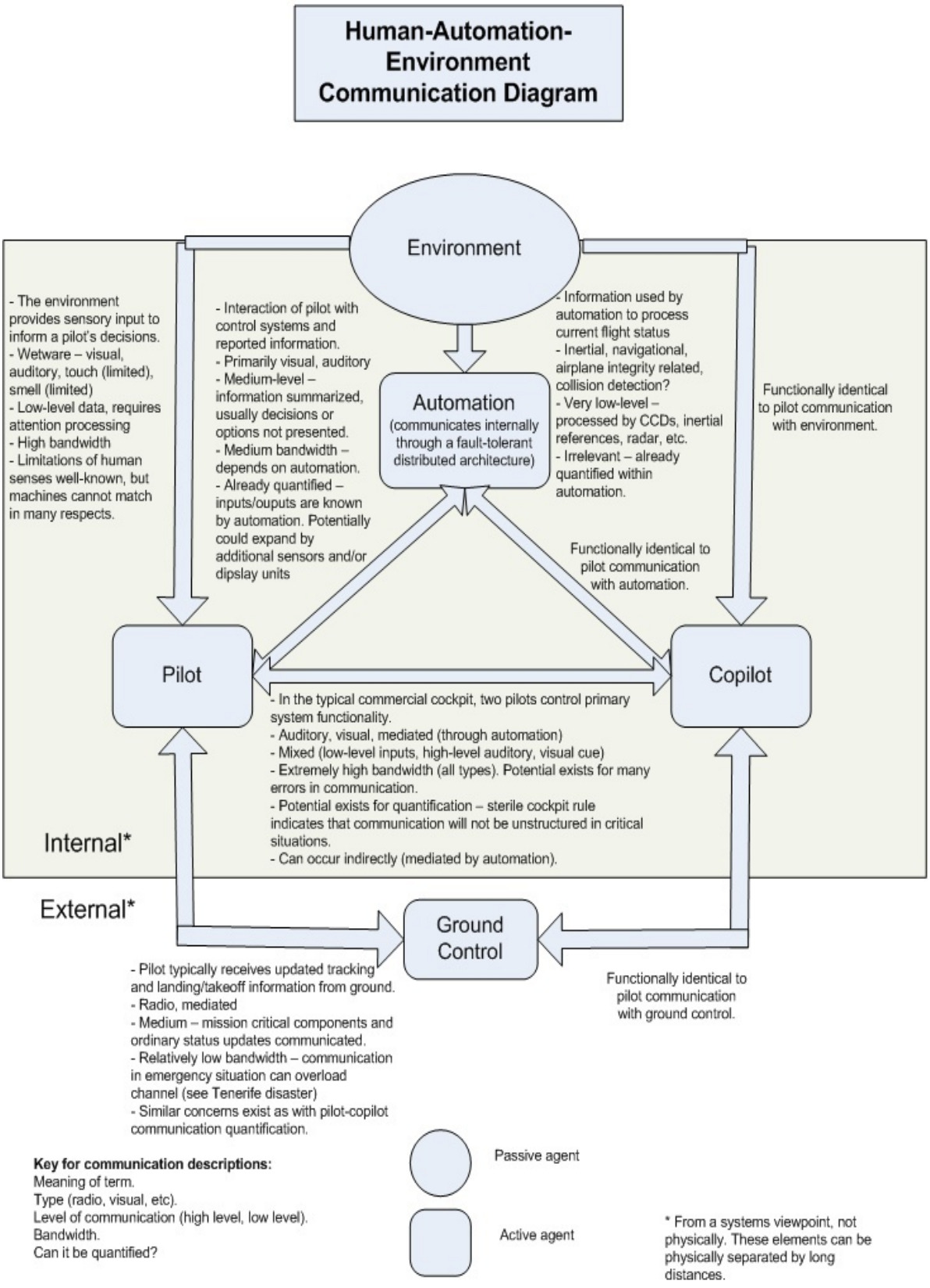
Analysis

The next three sections describe the bulk of our work so far, giving an overview of our research into human communications and failures as they relate to safety-critical systems in general and airplanes in particular.

Human communication model

As human communication in aircraft currently exists, several participants communicate, each with their own methods of communication, bandwidth, and relevant information to be shared. Generally, these participants include a pilot, a copilot, ground control, and the aircraft automation. The environment near the aircraft acts as a source of information that informs the decisions made by both pilots and the automation, based on their respective sensors as described in a later section. Figure 1 describes the basic structure of human communications at a link level. Ultimately, any system implementing a safe method of dynamic human/automation function allocation must verify that the necessary information communicated in figure 1 is still communicated in any situation where human responsibility is lessened. If information cannot be quantified, it cannot be analyzed within the task allocation system. A parallel effort to our project exists within NASA to improve sensor technology, awareness of pilot state, and aviation automation capabilities[YQ05]. Our analysis also indicates that such quantification will be made easier by the highly regulated and procedural nature of pilot-copilot-ground communications in emergency situations, the so-called “sterile cockpit rule”.

Figure 1 - Overview of Communications



However, we have some reservations about full quantification of communication within the cockpit. There does exist a significant amount of information not currently captured by FAA cockpit procedures, including facial expressions, gestures, body language, and casual conversations held while not in a critical situation. As well, the pilots in a typical cockpit act as checks on each other - adding a layer of partial redundancy to the information requirements for any automated system. Of these considerations, the redundancy may prove the most worrisome to replicate for two main reasons. These are: the traditional importance of pilot-copilot checks in flight operations and the individual differences between different flight crews in how and what checks are done.

The pilot-control communication links seemingly have much less unquantified extra information than the pilot-copilot connection, mostly due to the significantly decreased bandwidth of typical radio communications and strict regulations employed for such communication. For exactly these reasons, some aircraft are now moving toward a more data-oriented link, where telemetry and other information is communicated directly between ground control and flight computers. For the foreseeable future, however, the existing verbal link will not be eliminated. Another factor that must be considered is that, in the complete absence of human comprehension, the aircraft system also loses the ability to gain relevant information regarding the location of other aircraft relative to itself from their radio communications with ground.

Although some important information remains uncaptured by current automation, we remain confident that a model can be developed that addresses such issues to an appropriate level of satisfaction, based on significant human factors and sensors research in the area which should prune the remaining unquantified communications pathways, solving the problems of those relevant to the safety of the aircraft. Some questions do remain regarding human communication that exist on a higher level than our model. Do humans propagate failure between themselves? Related to this, what asymmetries in communication exist due to failure, and which exist due to the inherent communications protocol? How many fundamentally distinct communication graphs, in the computer science sense, are there? These questions must be addressed before formalization can proceed.

Human failure model

In parallel to our development of a model of human communications, we also considered the necessary criteria to establish a hierarchy of human misbehavior. This is a nontrivial task. The first question was to define a meaningful failure to base the rest of our theory on. We define a *meaningful failure* to be one that violates safety preconditions, in the aerospace sense. The question then must be asked, is this sufficient evidence to preempt human control? In general, it seems like there should be a very high level of evidence needed to preempt human control - anything lower may lead to distrust and fear of the system. It does not seem like the system diagnosis, fault categories, and error checks of ROBUS or other such distributed architectures can be directly applied. The sanity checks used for automation also cannot be immediately applied. For instance, humans are known for their self-correcting nature, but automation is generally not as was seen where a failure due to accumulated error in the internal clock of a Patriot missile battery in Dhaharan, Saudi Arabia let a Scud missile hit U.S. soldiers[8]. Therefore, any accusations of loss of control brought against humans may be irrelevant within a short time period. As well, humans operate on a vastly different timescale than automation, leading to synchronization issues.

[8]See `http://www.fas.org/spp/starwars/gao/im92026.htm` for more information.

Table 1: Failure Hierarchy

Hazard Scenario	SA Deficit	Symptoms	Feasibility of Recognition	Automation Response
1	Loss of all 3 levels.	Extreme exhaustion, loss of consciousness, physical incapacitation, death.	High – can detect loss of cabin pressure, high G forces, as is. Exhaustion detection is on IIFD road-map.	Take a safe action appropriate to the current flight pattern. Autopilot is in complete control.
2	Loss of recognition of critical factors in environment (level 1).	Inattention, exhaustion, ignoring critical information.	Medium – Unsafe and contradictory inputs can be caught at the bus level vs. the autopilot via an agreement algorithm.	Reject inputs and alert pilots as to situation. If leads to unsafe situation, preempt control temporarily (partially or fully – landing gear down, etc).
3	Loss of comprehension of current situation (level 2).	Impaired judgment, flight settings erratic.	Medium – Lack of inputs or overall violation of flight safety may be caught via an agreement algorithm.	Reject inputs and alert as to situation. Can lead to level 2 misbehavior and response if continued.
4	Loss of future perception (level 3).	Impaired judgment, flight settings erratic, continued deviation from scheduled flight plan and parameters.	Medium – how to determine impairment (alcohol, drugs), how to determine unsafe course of action? Long term deviation may be easier to detect.	Continue to monitor. If unsafe behavior becomes critical, preempt control.
5	None (normal operation)	None.	Not needed – continued, regular, correct following of flight plan and immediate control.	Take no action.

Of course, any combination of hazard scenarios 2, 3, and 4 can occur with lesser or greater severity. Hazard scenario 1 was described in detail because of its extreme nature and criticality – most likely to be the immediate application of dynamic function shifting.

To address these concerns, we look to human factors. The idea of the loss of situational awareness provides an attractive framework to classify human failures thanks to extensive research in the area and its direct relationship to potentially hazardous scenarios. As another benefit, situational awareness can be measured quantitatively using a variety of methods, including a standard test commonly used in human factors research[End88]. In this view, any lapse of control by the pilot can be explained as some loss of situational awareness. Even unconsciousness can be described as an extreme loss of all 3 levels of SA. With this idea in mind, we established the failure hierarchy shown in table 1, where categories are distinguished by the SA level loss experienced by the pilot.

Situational awareness is somewhat orthogonal to the idea of hazardous awareness states. Hazardous awareness states describe the cause of a pilot's state that may cause a potentially dangerous situation, while SA describes the effective state of the pilot's awareness itself. The three identified key hazardous states - preoccupation, vigilance, and absorption - do not seem to map cleanly to levels of SA. More work should be done on establishing a correspondence between these ideas. However, they do seem to have the same amount of power in classifying human failure scenarios, due to their general applicability. It seems as if the idea of hazardous awareness states lends itself more to actual physiological measurement, which has actually been carried out[PBB95]. This is

as opposed to SA, which is generally measured from external observation and testing.

We believe we have established a sufficiently broad model of human failure to begin formal analysis. It remains to be seen whether quantitative measurement of levels of SA can be effectively carried out, but based on previous work, we are optimistic. In our model, preemption is always carried out in the most extreme scenarios and never otherwise, as was intended. Some questions still remain unanswered. Our classification of human misbehavior is not yet at a level that can be evaluated in terms of the hybrid failure model from distributed systems. We are still unaware of where the integration of human misbehavior control and automation redundancy management would occur in the interaction of the two systems. Much work remains to be done.

Relevant differences between humans and automation

The previous sections have illustrated the need for a deeper understanding of the nature of the sensorial and computational differences between humans and automation. This understanding is needed to ensure that the automation can prove to take on the necessary roles of pilot in the case of an emergency.

Human beings have 5 distinct senses: smell, touch, taste, sight, and hearing. Taste is not relevant to airplane control and so was dismissed from consideration. The other four have some application in the cockpit. For example, smell can be used to detect changes in the atmosphere or burning in the aircraft. Touch can be used to detect unnatural vibrations within the aircraft potentially caused by malfunctioning engines. Hearing can aid in ascertaining the status of the aircraft and is used in all verbal communication. Sight is the most important of the senses to flying an aircraft and is used in almost all aspects of control and in maintaining situational awareness. Research in providing automation with these capabilities is at various stages of advancement, with computer vision probably being the most advanced, followed by auditory comprehension. For the most part, however, the differences between human and automation sensors seem to be a function of technological limitations, rather than a theoretical limit on automation capabilities.

Mental comprehension will be the hardest aspect of flight for automation to replicate. Developing a gestalt of a dynamic and unpredictable situation is currently an extremely difficult task for the automated reasoning present in aircraft systems. However, this high-level comprehension may prove to be unnecessary to substantially increase safety, as the NTSB reports we have reviewed indicate that small errors missed by the human pilot in emergency situations might have prevented major accidents. These low-level errors are especially appropriate for correction by automation due to its inexhaustible ability to perform a comprehensive number of system checks. As it has in almost all other functions of an aircraft, automation is likely to increase safety by being involved in command decisions.

Results

The subsequent two sections describe preliminary conclusions drawn from the analysis given above in terms of self-evident principles and mathematical properties. They are much easier to translate into the language of a formal verification system to verify or specify a particular safety-aware architecture than the previous analyses. Of course, there may be more lessons to learn from our work that have not yet been identified.

Principles of human safety

Two identified key principles found in our work are given below, with justifications coming immediately after. They are derived from our model of human communications in the cockpit and are most applicable to designing a specific architecture or family of architectures with particular fault assumptions in mind. They assume a bus architecture similar to that found in SPIDER.

1. *An active star bus topology is appropriate for a dynamic function allocation architecture.*
2. *Any message or vote regarding dynamic function allocation should take priority in the diagnostic schedule over all other ordinary bus diagnostics.*

Justification

The *active star* configuration entails any participating system node transmitting its message to all other appropriate recipients. This ensures that the maximum number of nodes are involved in any dynamic function allocation decision and receive information directly from source nodes. These advantages are vital when determining interactive consistency. This topology is used in ROBUS with the added complexities of bus interface units (BIUs) and redundancy management units (RMUs). It is even more important when dealing with information received from appropriate human state awareness sensors that should be combined by all available system nodes to make an informed decision.

Within a distributed architecture, a *communications schedule* is a priority ordering understood by the bus management system that gives the most important communicated information first (in either time or by some sort of priority signal) to a receiving node. Information regarding dynamic function allocation should be relatively infrequent in the bus due to the slower timescale of human actions versus that of bus communications. It needs to be acted on quickly in order to maintain system safety by ensuring that, regardless of the time of signaling, the first decisions made after a signal will be related to dynamic function allocation.

Safety/liveness properties for humans

The following two properties were derived from both our theoretical knowledge of safety/liveness properties from fault-tolerant design and our understanding of human factors work on aviation awareness. They are most appropriately applied in verifying a particular architecture or family of architectures, given appropriate fault assumptions.

1. *A system shall never be without a decision-maker for any designated safety-critical function.*
2. *Interactive consistency of judgments regarding pilot state will always be reached.*

Justification

The first item described here is a safety property, in the theoretical sense, describing the overall system goal for dynamic function allocation. The fundamental limitation of previously designed fault-tolerant architectures is leaving a hole where decision-making lies, with control issues decided entirely by the pilot. The fulfillment of the second item (a liveness property) is necessary

to the fulfillment of the first. In essence, the second property outlines the need for some type of agreement on the results of pilot state measurement before any decision-making can begin. The need for interactive consistency ensures that failures within the bus itself will not affect the function allocation process.

Decision-making on dynamic function allocation itself depends on developing an appropriate algorithm based on a failure model and the specific sensors involved. Although we believe we have created a sufficiently broad model of human misbehavior, the limitations of our understanding on potential sensor technologies prevent us from giving basic, essential principles for decision-making. At this time, we are unable to do so because we do not fully comprehend the relative importance of physiological and psychological measurements to our fault model. Developing this understanding should be the focus of continued work in this direction.

Future Work

This research has opened the door on at least as many questions as it has answered. The models of human communication and faults still need to be quantized into appropriate mathematical models that can be used to reinforce the justifications for our principles and properties. This approach will probably take a combined set/graph-theoretic approach with layers based on current sensor research on operator state as researched in other NASA projects[YQ05]. Many questions still remain about details of our models and their accuracy. An especially important consideration is how to establish true redundancy of the pilot by automation. The models need review by qualified persons in both the human factors or aviation communities and by the pilots themselves. When establishing fundamental principles to be used in future research, it is desirable that all of the implicit assumptions made in the models are discovered and enumerated.

After our principles are formalized and validated, it will finally become relevant and useful to model them formally using our experience with theorem-proving gained from SPIDER. Ultimately, the goal of our work remains to develop an architecture employing algorithms that can infer these properties. Whether such a prospect is within our theoretical grasp is not clear at this stage in the work, but one statement rings true: our goal will not be easy to achieve.

Conclusion

Due to the broad nature of the topic under consideration and our limited time to pursue said topic, with formal analysis being one of the most exacting and intensive techniques that can be performed on a system, our work has necessarily been an overview of topics that should be studied in more depth. It is true that we have developed preliminary models of human communication and a basic human fault hierarchy, along with proposed fault principles to match. This is only the beginning: there remain many avenues of research to be pursued. These include examining our fault principles, modeling communication in more depth, committing ourselves to basic sensors research, undertaking feasibility studies, and understanding policy-level issues. We have certainly unearthed many more puzzling facts than we have explained, but that is to be expected in fundamental research dealing with quantizing the vagaries of the human mind.

The feasibility of a safe and reliable realization of our work is still in question. But the work

itself - supporting transfer of duties between human and automation - is clearly important enough to pursue to see where it leads. Our work is preliminary and will most likely not be seen in commercial aviation for several years (if at all). However, it has the potential to save lives and provide for safer, more dependable air transportation. Our task is pursued in the best spirit of NASA's ongoing mission to research and develop aeronautical technologies for safe and reliable aviation systems.

Bibliography

[AAS98] Audit Report: Advance Automation System. Technical Report AV-1998-113, USA Department of Transportation, Office of Inspector General, USA, April 1998.

[AK00] Mohammad H. Azadmanesh and Roger M. Kieckhafer. Exploiting omissive faults in synchronous approximate agreement. *IEEE Transactions on Computers*, 49:1031–1042, 2000.

[ALRL04] Algirdas Avizienis, Jean-Claude Laprie, Brian Randell, and Carl Landwehr. Basic concepts and taxonomy of dependable and secure computing. *IEEE Transactions on Dependable and Secure Computing*, 1:11–33, 2004.

[EFJ+98] Mica Endsley, Todd Farley, William Jones, Alan Midkiff, and R. John Hansman. Situation awareness information requirements for commercial airline pilots. Technical report, International Center for Air Transporation, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, MA, USA, 1998.

[End88] Mica Endsley. Situational awareness global assessment technique (SAGAT). volume 3, pages 789–795. Proceedings of the National IEEE Aerospace and Electronics Conference, 1988.

[End95a] Mica Endsley. A taxonomy of situation awareness errors. In R. Fuller, N. Johnston, and N. McDonald, editors, *Human Factors in Aviation Operations*, pages 287–292. Ashgate Publishing Ltd., Aldershot, England, 1995.

[End95b] Mica Endsley. Towards a theory of situation awareness in dynamic systems. *Human Factors*, 37:32–64, 1995.

[JJG02] Jonas Jansson, Jonas Johansson, and Fredrik Gustafsson. Decision making for collision avoidance systems. Number 2002-01-0403, 2002.

[Lic60] J.C.R. Licklider. Man-computer symbiosis. *IRE Transactions on Human Factors in Electronics*, 1:4–11, 1960.

[LSP82] Leslie Lamport, Robert Shostak, and Marshall Pease. The Byzantine generals problem. *ACM Trans. Program. Lang. Syst.*, 4(3):382–401, 1982.

[NC00] Gleb Naumovich and Lori A. Clarke. Classifying properties: an alternative to the safety-liveness classification. In *SIGSOFT '00/FSE-8: Proceedings of the 8th ACM*

SIGSOFT international symposium on Foundations of software engineering, pages 159–168, New York, NY, USA, 2000. ACM Press.

[PB92] Alan T. Pope and Edward H. Bogart. Identification of hazardous awareness states in monitoring environments. SAE International Conference on Environmental Systems, 1992.

[PBB95] Alan T. Pope, Edward H. Bogart, and Debbie S. Bartolome. Biocybernetic system evaluates indices of operator engagement in automated task. *Biological Psychology*, 40:187–195, 1995.

[TPMM05] Wilfredo Torres-Pomales, Mahyar R. Malekpour, and Paul S. Miner. ROBUS-2: A Fault-Tolerant Broadcast Communications System. Technical Report TM-2005-213540, NASA, March 2005.

[Web06] Paul J. Weber. *Dynamic Reduction Algorithms For Fault Tolerant Convergent Voting With Hybrid Faults*. PhD thesis, Michigan Technological University, 2006.

[YQ05] Steven D. Young and Leighton Quon. Aviation Safety Program: Integrated Intelligent Flight Deck Technical Plan Summary. Technical report, NASA, 2005. Accessed from `http://www.aero-space.nasa.gov/nra_pdf/iifd_tech_plan_c1.pdf`.