Requirement Generation for Highly Integrated Aircraft Systems through STPA: an Application

Andrea Scarinci[1],
Massachusetts Institute of Technology, Cambridge, MA, 02139, United States

Amanda Quilici[2], Danilo Ribeiro[3], Felipe Oliveira[4], Daniel Patrick[5]
Embraer, São Jose dos Campos, SP, 12227-901, Brazil

and Nancy G. Leveson[6]
Massachusetts Institute of Technology, Cambridge, MA, 02139, United States

Abstract – *This paper presents an approach to requirement generation for complex and highly integrated aircraft systems using STPA, a hazard analysis technique that handles hardware, software, human operators and integrates them in a unified process. The approach is illustrated using the interfaces of the Air Management System (Engine Bleed, Cabin Air Conditioning, Pressurization and Anti-Ice) of a generic commercial aircraft. STPA is applied first to identify undesired/unsafe system behaviors through a structured, top-down approach. Requirements are subsequently generated from the results of STPA in order to handle these unsafe behaviors. Results from the application show that this approach allows systematically assessing the design space of the system starting from an early development phase and generating requirements to handle those properties that emerge from indirect component interaction and that often jeopardize the fulfillment of the high-level system goals. Human-machine interactions are also particularly well addressed by this methodology, integrating the human-factors process into the overall engineering process.*

1- Introduction and motivation

Today, aircraft systems are becoming increasingly complex, especially in terms of the high level of functional integration. Extensive use of software and automation has radically changed the way system components interact among each other [1]. Traditional functional decomposition frameworks that allowed describing the aircraft as a system composed of a number of almost-independent sub-systems (each with its allocated functions) are no longer sufficient to account for indirect interactions among aircraft components.

An example is given by the evolution of flight control systems. Once only composed of mechanical components (cables, turnbuckles etc.), they evolved into sophisticated servo-hydro mechanical systems and then transitioned to fly-by-wire technology. This last step has allowed the introduction of extensive automation (e.g. autopilots), which required adding an abstract architecture of the system to the physical architecture. The abstract architecture is defined by flight control laws and other models to represent the "desired" behavior of the aircraft. In the functions performed by the autopilot of a modern commercial airliner, the amount of information exchanged (input-outputs) with other systems is extremely high (e.g. air-data, landing gear, propulsion, avionics etc.) compared to the past. This created designs with intricate connections among components of sometimes completely different systems from a functional point of view (e.g. the possibility to

[1] Research Assistant, Aeronautics and Astronautics, 77 Massachusetts Avenue, Cambridge, MA, 02139, United States.
[2] Systems Integration Engineer, Embraer, 2170 Brigadeiro Faria Lima Ave, São José dos Campos, SP, 12227-901, Brazil.
[3] Systems Integration Engineer, Embraer, 2170 Brigadeiro Faria Lima Ave, São José dos Campos, SP, 12227-901, Brazil.
[4] Systems Integration and Safety Engineer, Embraer, 2170 Brigadeiro Faria Lima Ave, São José dos Campos, SP, 12227-901, Brazil.
[5] Product System Engineer, Embraer, 2170 Brigadeiro Faria Lima Ave, São José dos Campos, SP, 12227-901, Brazil.
[6] Professor, Aeronautics and Astronautics, 77 Massachusetts Avenue, Cambridge, MA, 02139, United States.

deploy the thrust reverser is determined by a vetoing logic that depends on: aircraft speed, monitored by the air-data system; measured weight-on-wheels, acquired by the landing gears; spoilers position, provided by the flight-control system etc.). Unexpected and unintended behaviors "emerging" from these kinds of interactions are often dangerous or simply not desirable and therefore need to be identified and handled as soon as possible during the design phase. An example is given by the accident that occurred to TAM Flight 3054, where unsafe interactions between the control logics of the throttle the propulsion control system, and possible deficiencies on behalf of the crew, resulted in a fatal crash [2].

One reason why these kinds of unexpected interactions continue to cause problems [3] is that they are often not identified during the conceptual design phase, when it is easiest and least expensive to eliminate. The way interfaces are usually documented is through an IRD (Interface Requirement Document), which contains detailed descriptions of the hardware interfaces and a list of the information/messages exchanged: the functional aspect is often not addressed in a direct way. Functional decomposition frameworks lead the analyst to consider each function separately and make it difficult to identify unintended interactions between components fulfilling different functions. Traditional hazard analysis techniques also only partially address this problem in the sense that they tend to focus on hardware reliability and often assume independence between components or subsystems and direct interactions [1, 4].

In this paper, the application of a methodology is presented to identify and analyze the interfaces of the Air Management System (AMS) of a generic commercial airliner and to generate the system and component-level safety requirements. First, a hazard analysis technique called *System Theoretic Process Analysis (STPA)* [5] is applied to identify unsafe system behaviors and the causal scenarios that can lead to them. STPA employs a systemic approach to architecture/functional analysis that includes the search for indirect interactions between components. STPA has an underlying theoretical foundation in system theory and therefore frames the engineering problem as a control problem, uses a feedback control model as a basis for system modeling, and hierarchy and abstraction levels as a means of managing complexity. The human operator and his/her interactions within the system are included directly in the modeling and analysis in the same way that hardware and software are, i.e., as just another system component.

The results of the application in this paper show that STPA successfully captures emergent behaviors due to inter-system component interactions through a structured, systematic process. Once these behaviors are identified, it is possible to generate appropriate requirements to handle them and design features to ensure them. STPA therefore provides the basis for requirement cascading and refinement with traceability and a robust rationale associated. More discussion on the application of the technique as well as the results found in the specific system analyzed in this paper are presented in section 3 and 4.

DISCLAIMER: The results presented in this paper are the end-product of a pilot project performed at Embraer (Brazil) and all the information presented is to be considered for illustrative purposes only.

2 - Application to the Air Management System (AMS) of a generic commercial airliner

In this section, STPA is applied to analyze the interfaces of the Air Management System of a generic commercial aircraft. A brief description of the object of study is presented first, followed by the presentation of the application of the technique in the context of the AMS.

2.1 – Air Management System (AMS)

The system chosen for the application presented in this paper is the Air Management System of a generic airliner. The Air Management System is intended here as the system that manages air extraction from the engines, Auxiliary Power Unit (APU) and/or the external environment for cabin and cockpit air conditioning, aircraft pressurization and anti-ice accumulation on the wing as well as other areas of the aircraft. As previously announced, the scope of this analysis is to identify potentially undesired interactions between the AMS and its interface systems. In this sense, the AMS is a particularly interesting system to analyze as it interacts with numerous other systems including the electric system, propulsion system, hydraulic system/landing gear, fuel-system, flight controls system, avionics as well as the cabin interior/aircraft structures. The AMS is treated in this application as an entity with both hardware and software features exchanging signals and commands both internally and externally (i.e. it is not just its black-box behavior that is being evaluated e.g. Table 9 – UCA 30).

The interactions with the propulsion system are particularly significant. The AMS has to regulate the air flux extracted from the engines according to its needs as well as thrust-level requirements to guarantee satisfactory performance. It is of vital importance that the propulsion system main function (providing forward thrust) is not jeopardized due to incorrect balancing with the air-conditioning needs or as a consequence of minor failures within the AMS itself. A particularly dangerous case is that of reverse flux from the AMS to the engine. At the same time, satisfaction of cabin air pressure, temperature and quality requirements must also be guaranteed. The other important interface of the AMS is the cabin environment. Maintaining adequate temperature and pressure in the various environments of the aircraft (with continuously changing thermal loads and external environmental conditions) is a challenging task. Location of sensors and transients need specific attention.

The rest of the interfaces of the AMS are somewhat less extensive. The interactions with the electrical system are mainly limited to power supply management. The interactions with the hydraulic system/landing gear are limited to the exchange of the ground/flight information (which is important in some of the software logic that regulates cabin air management). The interactions with the fuel system concern the cases in which an inerting system[7] is installed. The AMS also interacts with the Flight Control System (FCS) as it heats the slats to avoid ice accumulation. As mentioned earlier, besides the hardware parts of the system, the AMS also includes sophisticated software that largely interacts with the avionics system: it handles inputs coming from the cockpit (e.g. temperature settings) and acts as an information sharing node with other subsystems. In the analysis presented in this paper, human interactions with the AMS have also been included. Cockpit and cabin crew together with maintenance personnel have been taken into consideration. A special role has also been reserved for how passengers could interact with the cabin environment in a way that the AMS could be impacted. These kinds of interactions are often assumed to be nominal during the system design, which, however, is often not the case during operations.

As can be seen more specifically in section 3, the level of detail of the analysis is quite low, i.e. the features of the AMS here considered are pretty generic and apply to almost every commercial aircraft and can be identified during early system analysis and concept development. Significant and important requirements can be generated even at this early stage of development through the process described in the following subsection. As development proceeds, the generic requirements can be refined into more detailed requirements and design decisions as shown in the AMS example in this paper.

2.2 – STPA and requirement generation

As mentioned in the introduction, the approach to requirement generation consists of using STPA (a relatively new hazard analysis technique) to identify potentially undesired/unsafe scenarios and then the identification of system and component requirements to adequately address these cases. This section describes the steps in the proposed process and related A.M.S. application.

2.2.1 – STPA

Accidents and hazards

The first step in any STPA analysis is to identify what the system is "not supposed to do", i.e., identify the accidents and hazards associated with the system. The general definition of both accidents and hazards used are [6]:

Accident: "An undesired and unplanned event that results in a loss, including a loss of human life or human injury, property damage, environmental pollution, mission loss, financial loss, etc."

Hazard: "A system state or set of conditions that together with a worst-case set of environmental conditions will lead to an accident (loss)"

An accident can therefore be loss of human life, damage to physical equipment, an economic loss or the mission objective not being achieved. The accident definition used by STPA is purposely broadened with respect to the ICAO[8] or FAA[9] definitions. A more limited definition of accident and hazard can be used, but the results will also have more limited usefulness. The broad definition used here is essentially that used for military aircraft in MIL-STD-882.

---

[7] An *inerting* system tries to keep a chemically non-reactive gas-environment inside the fuel tanks.

[8] ICAO accident definition: "An occurrence associated with the operation of an aircraft which takes place between the time any person boards the aircraft with the intention of flight until such time as all such persons have disembarked, in which: a) a person is fatally or seriously injured [...] b) the aircraft sustains damage or structural failure [...] c) the aircraft is missing or is completely inaccessible."- *Aircraft Accident and Incident Investigation – Annex 13 – ICAO.*

[9] FAA accident definition: "Aircraft accident means an occurrence associated with the operation of an aircraft which takes place between the time any person boards the aircraft with the intention of flight and all such persons have disembarked,

A hazard represents a state of the system for which some worst-case conditions can exist that can lead to an accident. Examples of hazards include excessive vertical speed, engine stall, and loss of control. Depending on the particular conditions (e.g. current aircraft altitude, weather conditions, experience of the pilot flying etc.) these system states could eventually evolve into a loss. STPA is based on worst case analysis, not "average" or "expected" case analysis. If there are no conditions under which an accident could occur when the aircraft is in some particular state (no worst case exists), then there is no problem and thus no hazard.

The ideal operational scenario is one that precludes the system from entering a hazardous state. STPA allows systematically exploring the design space of the system in search of those conditions that can lead to potentially hazardous system states. The objective is to establish *safety constraints* to ensure operations always are within the desired safe perimeter (Figure 1).

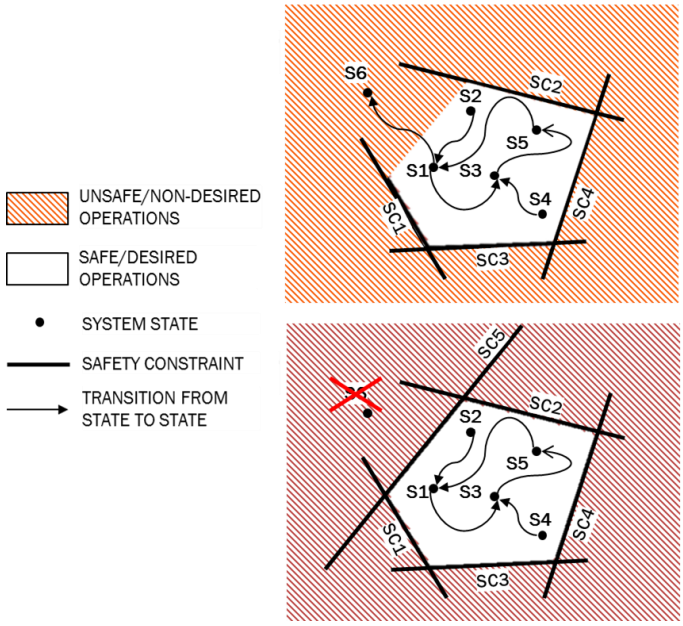

Figure 1 – Safety Constraints and System Operations

For the specific AMS application the following hazards and accidents have been identified

Accidents:

A-1: Loss of Life / Injury (suffocation, eye/ear irritation etc.);
A-2: Loss/damage to aircraft and its equipment
A-3: Mission Interruption or delay

Hazards:

H1: High/Low Air Temperature
H2: High/Low Air Pressure
H3: Inappropriate Air Transport (bleed and distribution)
H4: Unacceptable Air Contamination
H5: H_2O/Ice (other) Accumulation

These five hazards are related to the functions that need to be performed by the AMS system. The temperature of the cabin air and other environments must be kept within an appropriate range (H1). The same is true for cabin and duct pressure (H2). The transport of air from the engine to the various areas of the aircraft, as well as exchanges with the external environment must be performed without losses, with adequate flow rate and to/from the appropriate locations (H3). Air must also be kept "clear" i.e. not contaminated (H4). The AMS must also assure water and/or ice is not accumulated on specific surfaces (H5). This function could be included in that of maintaining the air at an adequate

---

and in which any person suffers death or serious injury, or in which the aircraft receives substantial damage. "- *Title 49 CFR 830.2*

temperature (classical anti-ice system for wing leading edges). However, in some aircraft, the anti-ice function is also performed through electrical heating and windshield wipers are sometimes considered part of the AMS system as well.

The high level of these definitions is particularly important to ensure that the list of hazards is as comprehensive as possible and can be adapted to the specific exigencies of each aircraft manufacturer. As an example, Hazard 2 includes any condition in which the air in a given environment is not maintained at the adequate pressure. The most typical case would be that of a drop in cabin pressure (loss of cabin pressurization). The reason why the specific term cabin is not mentioned is that there is not just the cabin air that needs to be maintained at adequate pressure. Pressure in the air ducts needs to be controlled, pressure for engine start etc. If H2 were to be replaced with more detailed hazards, the list would be very long and some specific condition will most certainly be forgotten. In other words, it is easier to recognize a specific critical condition by associating the words "high/low pressure" to each control action, rather than building more detailed hazards from the start.

These hazards can easily be transformed into related safety constraints (or design constraints). These constraints are fundamental as they define the *rationale* for the more detailed requirements that will be generated during the process and define the goals of the engineering design process:

SC1: The AMS must not let the air temperature reach values out of the prescribed limits for the destination environment [→ H1];
SC2: The AMS must not let the air pressure reach values out of the prescribed limits for the destination environment [→ H2];
SC3: The AMS must not extract air from inappropriate sources at an inappropriate time [→ H3];
SC4: The AMS must not transport air to inappropriate environments at inappropriate times [→ H3];
SC5: The AMS must not distribute air inside the aircraft that is unacceptably contaminated [→ H4];
SC6: The AMS must avoid H_2O/Ice accumulation on specific surfaces [→ H5].

H3 has been split into two different safety constraints (SC4 and SC5) because, beyond issues related to air leaks or transport to the wrong part of the aircraft, a specific problem exists in terms of the way air bleed sources are managed. In particular, excessive demand from the engines must be avoided in order not to jeopardize engine performance (e.g. excessive air conditioning demand during take-off on a hot day).

More detailed safety constraints are generated during the identification of unsafe control actions (see unsafe control actions identification section).

Control Structure

STPA is based on systems theory and treats safety as a *control problem*: the controller (human operator, computer, organization or institution, etc.) needs to ensure that the safety constraints are never violated, that is, the process behaves safely during operations. A process can be a physical process (e.g. set of actuators, a turbine etc.), but also a computer, an organization or a mixed system (e.g. air traffic control made of pilots, aircraft and ground controllers).

In order to define the safety constraints needed to avoid the identified hazards, STPA treats the problem as one of control and a model is first created of the system as a hierarchical control structure made up of simple feedback control loops. Figure 2 shows the general form of a simple feedback control loop.

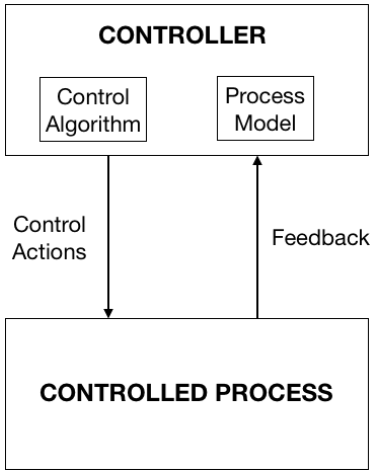

Figure 2 – Simple Feedback Control Loop (adapted from [7])

The ways in which the controller can change the state of the process it is controlling are called *control actions* (e.g. a pilot pulls the yoke and commands a change in position of the flight surfaces, a computer commands the opening of a valve etc.). The controller also needs to receive updated information on the current process status through appropriate feedback (e.g. sensors, visual monitoring, auditing, etc.). This feedback information is used to update the internal process model. Every controller needs a model or information about the state of the system it is controlling in order to provide appropriate control actions. In humans, we usually use the term "mental model" instead of process model.

A controller makes decisions about what is the correct control action to produce and when to produce it based on its embedded *control algorithm* and *process model*. A control algorithm is any kind of process (e.g. rules, procedures etc.) based on which a controller selects the actions to take (normally from a range of alternatives). In humans, the decision-making strategies can vary from individual to individual and according to context.

Control algorithms use the assumed current state of the controlled process (the process model) to support the decision-making process. Leveson [8] says: "the process model includes assumptions about how the controlled process operates and about the current state of the controlled process. Accidents in complex systems, particularly those related to software or human controllers, often result from inconsistencies between the model of the process used by the controller and the actual process state. The inconsistency contributes to the controller providing inadequate control". For example, in the Air France 447 accident the air data system was not working properly. Due to a number of inconsistent cues coming from the cockpit, inadequacies in the training process and other factors, the crew diagnosed the situation as an over-speed event, while in fact the aircraft was undergoing a low-speed stall. The pilots therefore undertook a series of actions to reduce speed, while also following inappropriate indications from the flight director, instead of applying what would have been the appropriate "doubtful air speed indication" procedure [9]. Another example is when the automation, due to a sensor failure and software logic design, detects the aircraft as "in-flight" while already on the runway and does not allow the pilots to activate the reverse thrust [10].

Based on its control algorithm, process model, control actions and feedback mechanisms, the controller needs to enforce the established safety constraints to ensure safe operations. The control loop model illustrated in Figure 2 is adaptable to almost any kind of engineered system and allows an integrated analysis of the human element together with the software and hardware portions of the system. Note that this model is very different than the common architectural model used in system engineering to show the interconnections and data flow between system components. Architectural connection models do not define functional control and the feedback control loops.

In STPA, the individual control loops are put together into a hierarchical control structure of the functional system design. By building a model of the control structure, the engineer is forced to think about which element is controlling what in the system and therefore to formalize the delegation of roles and responsibilities

In the A.M.S. case the control structure looks as follows: a first high-level control structure (with very few details included) is shown in Figure 3.

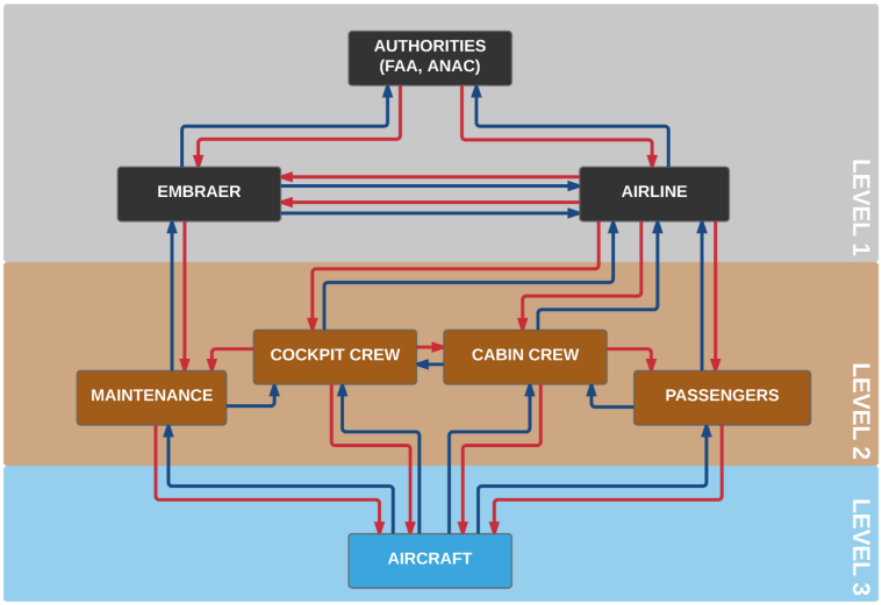

Figure 3 – High-Level Control Structure for Aircraft Level Analysis

There are three levels: level 1 (omitted from later analysis in this paper) includes entities such as aircraft manufacturer, governmental authorities and operators (e.g., airlines); level 2 focuses on the human operators directly involved in the system of study (pilots, cabin crew, passengers and maintenance personnel); and level 3 represents the aircraft.

In this paper, level 1 has been excluded from the analysis in order to reduce the example size. Factors such as the regulations imposed by the authorities, the manuals printed out by airliners and aircraft manufacturers, their supply chain for maintenance etc. can heavily impact the safety of operations. An accident could originate from a badly written procedure or lack of legislation. Teams primarily involved with the regulatory and certification aspects of the design will be concerned by this kind of analysis. However, also engineers devoted to manuals/procedure elaboration will have a primary role. Ultimately all systems specialists, maintenance experts and pilots must not forget that an aircraft is part of a bigger system that goes beyond the perimeter of the airframe itself.

A more detailed control structure has been produced for levels 2 and 3 (Figure 4). In this example, the objective is to analyze the interfaces of the AMS system. For this reason, the control structure focuses on the functional relation between the AMS and the other aircraft systems. The role of the operators is included: cockpit crew, cabin crew, passengers and maintenance personnel. Cabin crew and passengers have an active role as they can significantly interact with the cabin environment. Flight attendants can set cabin temperature or report problems. The passengers define the thermal load and can also interact with some of the devices that are part of the AMS (e.g. temperature sensors, smoke detectors, gaspers etc.).

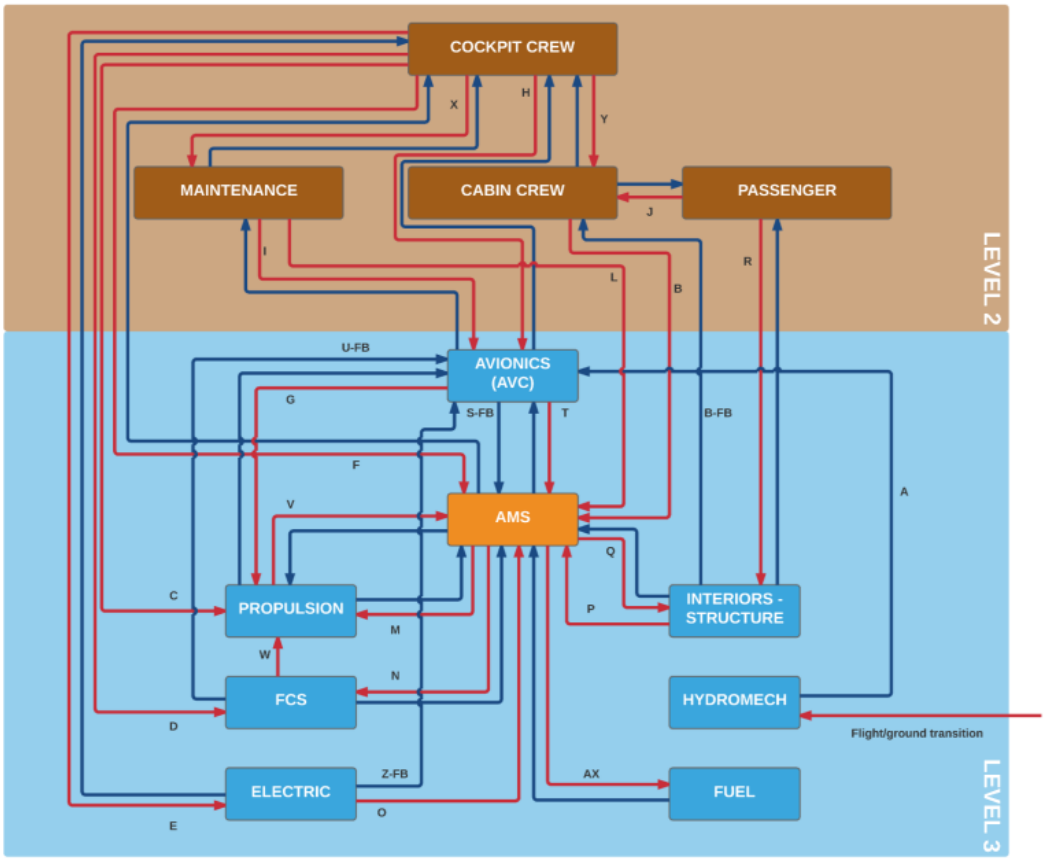

Figure 4 – Control Structure for the interfaces of the Air Management System

In Figure 4, the red arrows represent control actions, the blue ones represent feedback. The letters are just used as a reference to list all the control actions associated with the specific interface in a separate table for clarity purposes and to avoid too cluttered a figure. Feedback is referenced in the same way. An example is shown in Figure 5. Control actions are indicated with the reference letter (associated with a specific interface e.g. F for Cockpit-Crew A.M.S.) plus a number that identifies individual control actions that are part of the same interface. Feedback is distinguished from the control actions with the prefix FB. Examples of control actions are: pilot regulates the cabin or cockpit temperature (F7-F3), deactivates air recirculation or air conditioning unit (F9-F6) by means of buttons on the cockpit panels. These signals go directly to the AMS and the only feedback provided of the signal being sent is the push-button position (or feedback light). This, of

course, doesn't mean necessarily that the AMS has received the command or that it is implementing it even if it has received it. Some information about what is actually happening "in" the system can be physical and is also provided through the cockpit displays (H2-FB, T1-FB), however the pilot has to manually select the specific synopsis page to access this information (control action H2), which means the data may not always be directly visible to the crew. This way of representing the system (controller and controlled process) highlights the system functional architecture in terms of communication paths and clearly indicates which controller is sending/receiving the information and to/from where.

The level of abstraction of these control actions is, again, quite high. This is done on purpose in the effort to maintain completeness. As an example, the control action Q1 "Sends and distributes conditioned air" can seem vague and not very useful from an engineering point of view. However, while working at this level of detail does not allow evaluating the specific internal architecture of the AMS, it allows focusing on high-level interactions with other systems. As an example, sending conditioned air may be an issue when the air being sent is contaminated. It can also be a problem when there is a fire in some areas of the aircraft and the AMS keeps spreading smoke inside the cabin. Starting the analysis at this level before later focusing on details allows better identification of possible unsafe interactions that would be difficult to discern if working with more specific information. An even higher level of abstraction is contained in the control action A1 "Land/Take-Off": the objective here is not to analyze these operations, but to evaluate the impact of the take-off and landing phases on the AMS system. Temperature and, particularly, pressure changes must be handled carefully during these transitions (flight to ground and ground to flight) and defining this control action allows doing so. Redefining the control action A1 as "setting flaps to take-off configuration", "setting the thrust to TOGA", "rotation" etc. introduces a granularity that is not necessary to evaluate the impact on the A.M.S. at the level chosen. In fact, this more detailed control action list could introduce the possibility of missing the take-off scenario as none of the actions listed specifically relates to take-off and some of them, analyzed as individual U.C.A.s, don't have a specific impact on the A.M.S. at all (e.g. rotation).

Controller		Control Actions	Controlled process
Pilot	F7	Regulate Cockpit Temperature	A.M.S.
Pilot	F3	Regulate Cabin Temperature	A.M.S.
Pilot	F9	Deactivate Air Recirc	A.M.S.
Pilot	F6	Deactivate Air Conditioning Unit (Right, Left, Both)	A.M.S.
Maintenance	L1	Deactivate Air Conditioning Unit (Right, Left, Both)	A.M.S.
Pilot	F6-FB	FB: button	A.M.S.
Pilot	H2	Select Bleed Page on dedicated display	A.V.C.
Pilot	H2-FB	FB: A.M.S. status (pressure, temperature, valves etc.)	A.V.C.
A.V.C.	T1-FB	FB: A.M.S. pressure, temperature, valves values and status	A.M.S.
A.M.S.	Q1	Sends and distributes conditionned air (cabin, avionic bay etc.)	Interior
A.M.S.	Q1-FB	FB: temperature	Interior
A.M.S.	S2-FB	FB: external temperature, altitude etc. (from Air Data sub-sys)	A.V.C.
Cabin Crew	B1	Regulate Cabin Temperature	Interior
Cabin Crew	B1-FB	FB: Physical	Interior
Pilot	A1	Land/TakeOff	HYD
A.M.S.	S1-FB	FB: WoW (weight on wheels - boolean)	AVC
A.V.C.	A1-FB	FB: WoW (weight on wheels - boolean)	HYD
A.M.S.	M4	Ensure correct air flow direction	Propulsion System
A.M.S.	M4-FB	FB: none	Propulsion System

Figure 5 – Control action examples

Building the control structure allows starting the analysis of the system itself. It can be easily seen that in our case some control loops have no feedback (e.g. the electric system is not aware of whether the A.M.S. is receiving power or not) or the feedback follows a "different" path within the system compared to that of the associated control action (e.g. the pilot sends a thrust command through the throttle to the propulsion system, however the feedback data are handled and filtered through the avionics). These asymmetries in the flow of information can sometimes cause the system to behave in an undesired way and are often the source of those emergent behaviors that are difficult to identify through functional decomposition methods.

The concepts of control action and feedback can sometimes be adjusted to describe particular aspects of the system as it happens here with the flight/ground detection function performed (partially) by the landing gear. In this case, a decision has been made by the authors to consider the runway as the control agent. The runway, by compressing the landing gear and making wheels turn, induces the software logic to switch from flight to ground (or vice versa). These types of

assumptions are documented in the control structure and analysis so that changes can later be made if necessary and the analysis traced to the higher-level assumptions.

The control structure can become quite complex, even when a high level of abstraction is used. While it is impossible to guarantee 100% completeness, the control structure has been checked by multiple system designers, pilots, maintenance staff, each for his/her domain of knowledge. In general, it is also possible to refine the control structure during the analysis to add any missing information.

Concerning the A.M.S. application, the control actions analyzed in this paper are control actions M4 (Ensure correct air flow), L1 (Deactivate Air Conditioning Unit - maintenance) and F6 (Deactivate Air Conditioning Unit – pilots). For each of them the STPA analysis requires the identification of possible unsafe control actions and causal scenarios. Finally design recommendations are created with associated requirements.

Control action M4 – Ensure correct air flow direction

Unsafe Control Action Identification

There are four types of potentially unsafe control actions [11]:

- A control action required to prevent a hazard is not provided or not followed;
- A control action is provided that leads to a hazard;
- A potentially safe control action is provided too late, too early, or out of sequence, leading to a hazard;
- A potentially safe control action is stopped too soon (for a continuous or non-discrete control action) or applied too long, leading to a hazard.

The unsafe control actions can be documented in a table (see Table 1) where each row represents a possible control action (listed in column 1) as modeled in the hierarchical control structure and the columns represent the context in which the four types of unsafe control can occur. The content of the table is the context (conditions) under which that control action could lead to an unsafe or undesired outcome. It is important to note that the table does not just list failure conditions (which would be a bottom-up FMEA type process). Failures are not listed. Instead, the context in which the designed system control actions can lead to hazards is identified. This context will later be used to identify how the control action could occur in those unsafe contexts, that is, the causal scenarios for the hazards.

Control action M4 is related to a specific function that must be ensured by the AMS system: the air flow must always go from the engines or A.P.U.[10] to the aircraft interiors. “Reverse flow” must never occur to avoid engine shut down. Using this information, it is possible to identify the unsafe control actions in Table 1.

Table 1 – Unsafe Control Actions for Control Action M4

ID	Control Action	Provided	No.	Not Provided	No.	Too Late, Too Early, Wrong Order	No.	Too long, too short
M4	Ensure correct air flow direction		205	The AMS does not ensure the correct air flow direction when manifold pressure is higher than engine pressure (reverse flow) [H3-H2]	206	The AMS ensures the correct air flow direction too late when manifold pressure is higher than engine pressure (reverse flow) [H3-H2]	207	The AMS ensures the correct air flow direction for too short time when manifold pressure is higher than engine pressure (reverse flow) [H3-H2]

[10] A.P.U. = An auxiliary power unit is a small turbomachine used to provide air bleed for air conditioning and engine start (when the engine start is pneumatic) or/and electrical power when the engines are not running. When the engines are running it can also be used as a redundant source of electrical power and compressed air (mainly in critical phases of the flight such as takeoff and landing).

	208	The AMS does not ensure the correct air flow direction when manifold pressure is higher than APU pressure (reverse flow) [H3-H2]	209	The AMS ensures the correct air flow direction too late when manifold pressure is higher than APU pressure (reverse flow) [H3-H2]	210	The AMS ensures the correct air flow direction for too short time when manifold pressure is higher than APU pressure (reverse flow) [H3-H2]

UCAs 205 and 208 identify situations in which the correct control action is not provided i.e. the reverse flow is not prevented at all in one case towards the engine and the other case towards the A.P.U. The other UCAs deal with timing issues: a mechanism to prevent the reverse flow must be carefully regulated in order to be safe. Note that the unsafe control actions are solution independent. The functional needs are only being identified here. Whether this will translate into a system of valves or other solutions will depend on the specific choices made by the engineers to prevent the identified unsafe control actions. The generic architecture being dealt with here is shown in Figure 6. The only fixed elements are the engines, the A.P.U., ducts and the idea of some mechanisms to cut-off or add air sources. These elements are common to almost any type of commercial aircraft. Despite this level of generality, important basic requirements can still be generated.

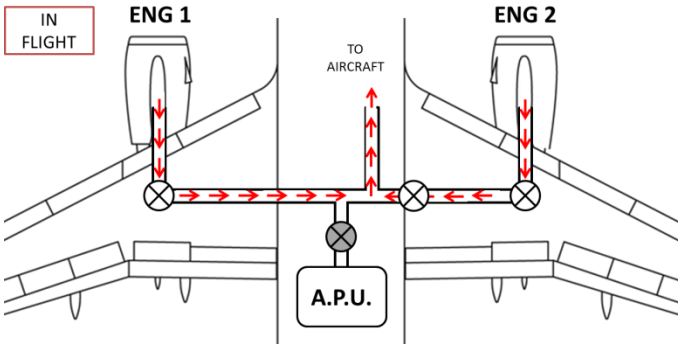

Figure 6 – Bleed Air System

At this point it is possible to expand safety constraint SC4 into more detailed constraints concerning this specific problem using the identified UCAs.

SC205 The AMS must not allow reverse flow when the manifold pressure is higher than the engine pressure. [→SC4→H3]

SC206 The AMS must not allow reverse flow to last for a period of time longer than that specified by the engine manufacturer. [→SC4→H3]

SC207 The AMS must guarantee reverse flow mechanisms are active for a long enough time to prevent the reverse flow when the manifold pressure is higher than the engine pressure. [→SC4→H3]

SC208 The AMS must not allow reverse flow when the manifold pressure is higher than the A.P.U. pressure. [→SC4→H3]

SC209 The AMS must not allow reverse flow to last for a period of time longer than that specified by the A.P.U. manufacturer. [→SC4→H3]

SC210 The AMS must guarantee anti-reverse-flow mechanisms are active for a long enough time to prevent the reverse flow when the manifold pressure is higher than the A.P.U. pressure. [→SC4→H3]

Causal Scenarios

The identification of unsafe control actions is only the beginning of the analysis process. The designers need to know the causal scenarios that can lead to unsafe control actions to have the information necessary to eliminate or mitigate unsafe control. Two types of scenarios can lead to a hazard: 1) unsafe control is provided; 2) safe control is provided by the controller but not implemented by the process (perhaps because of some physical component failure). Physical component failures are usually captured in traditional hazard analysis techniques, but those techniques miss the accidents that occur even when there are no physical failures. STPA captures both.

For the UCA206 illustrated above, the following general causal scenarios can be identified (Table 2):

Table 2 – Causal Scenarios for Unsafe Control Action 206

U.C.A.	**206**
The A.M.S ensures the correct air flow direction too late when manifold pressure is higher than engine pressure (reverse flow)	
Scenarios	
1	The AMS does not have a physical means to avoid the reverse flow
2	The AMS commands the engine bleed valves opening before the APU bleed valve is completely closed (a-synchronized command)

The first scenario may seem trivial, but it is important to make sure certain functionalities are linked to requirements with appropriate rationale. This scenario can include not only the absence of equipment to stop the reverse flow, but also sensors or a control algorithm and process model to be activated when needed. More details of the functional architecture may be included during later iterations of STPA.

The second scenario identifies a specific coordination problem (common to almost any commercial aircraft) that is often the cause of reverse flows. The A.P.U. can in many cases produce enough air pressure to counterbalance that coming from the engines (as a matter of fact the A.P.U. is used to start the engine, thus the air flux provided is quite significant). During phases in which the engine is at IDLE (lowest thrust level), this scenario becomes particularly likely. There are specific operational scenarios in which the pilot may switch the A.P.U. on well before the engines are shut down: some pilots switch the A.P.U. on before landing to anticipate air conditioning needs on the ground or to use it as redundant electrical power and/or bleed air in case of failure of the engines during this critical flight phase. However, during the approach/landing phase, the thrust can often be at IDLE or close to it, with the risk of causing reverse flows and engine shut down (Figure 7). For this reason, the engine bleed valve must never be opened before the A.P.U. bleed valve is closed and the A.P.U. valve never opened before the engine bleed valve is closed. Because this scenario relates to the engine, only the former case is specified under UCA 206. The scenario here exposed is an example of how a safety measure (redundancy) undertaken by the pilots can turn out to be unsafe when some interactions with other system components have not been adequately considered.

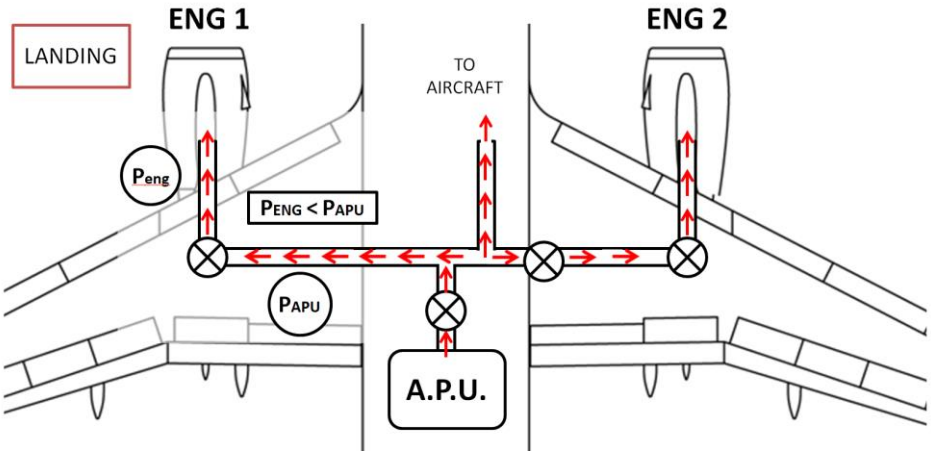

Figure 7 – Reverse flow during landing. Airframe schematics adapted from [12].

The choice has been made in this application to keep the level of abstraction of the scenarios quite high. More details will be provided through the requirements (see next subsection), while this part of the analysis remains generic enough to be re-used in other projects.

Design Recommendations and Requirements

The identification of possible hazardous scenarios provides information to the designers so they can make more informed decisions. In this subsection, the design recommendations generated for the identified scenarios are shown (
Table **3**).

Table 3 – Design Recommendations for UCA 206

No.		**SAFETY CONSTRAINTS**
206		The AMS shall not allow reverse flow transient in the air duct to last longer than that specified by the engine manufacturer (risk of engine shut down)
No.		**DESIGN RECOMMENDATIONS**
DR	206.1	The valves closing time shall be small enough to avoid engine shut down
DR	206.2	The software / analog device that controls non-purely-mechanical valves shall take into consideration the time it takes to operate them (e.g. time lag etc.)

Design recommendation 206.1 suggests paying specific attention to the closing time of the valves used to isolate the engine, while the second recommendation adds to the mechanical aspect of the complexities linked to automatic control. Not only will the valve have its own transient time, but the lag between when the signal is sent by the computer and received by the valve will impact the effectiveness of the solution used. At this point it is possible to generate specific requirements for the design recommendations just illustrated (Table 4). The requirements contain specific information about the solution chosen.

Table 4 – Requirements for UCA 206

Nbr		SAFETY CONSTRAINTS		
206		The AMS shall not allow reverse flow transient in the air duct last longer than what specified by the engine manufacturer (risk of engine shut down)		
Nbr		**DESIGN RECOMMENDATIONS**		
DR	206.1	The valves closing time shall be small enough to avoid engine shut down		
DR	206.2	The software that controls non-purely mechanical valves shall take into consideration the time to operate them (e.g. time lag etc.)		
Nbr		**REQUIREMENTS**	**TRACEABILITY TO DESIGN RECOMMENDATIONS**	**VERIFICATION METHOD**
R	206.1	The PRSOV valve shall close to 95% of its "fully-closed" position value in 0.2 seconds when the manifold pressure is higher than engine pressure,	DR206.1, DR206.2	Bench Test
R	206.2	The check valve shall close in 0.1 seconds in case of PRSOV failure in order to prevent the reverse flow	DR206.1	Bench Test
R	206.3	The AMS controller shall command the PRSOV valve to be closed at the same time as the HPRSOV	DR206.1, DR206.2	Rig Tests
R	206.4	The HPRSOV shall close to 95% of its "fully-closed" position value in 0.2 seconds after commanded to close	DR206.1, DR206.2	Bench Test

The names of the valves PRSOV (Pressure Regulating Shutoff Valve) and HPRSOV (High Pressure Regulating Shutoff Valve) refer to a specific architecture chosen to manage air bleed from the engine. Besides the valve names, however, it is important to note that these requirements are testable, complete and clear. One requirement (206.1) describes by how much and how quickly the PRSOV valve must be closed when the manifold pressure is higher than the engine pressure. The same is true for requirement 206.4 concerning the HPRSOV. Requirement 206.2 establishes the behavior of a check-valve, to be operated only in case of PRSOV failure. Requirement 206.3 directly covers scenario 2 for UCA 206. Of course, other design recommendations and requirements could be generated to address the issues identified through the scenarios of UCA206. What is considered important here, however, is the logical flow-down from the hazard to these low-level requirements. The results are summarized in Figure 8.

Figure 8 – Requirements Traceability

Traceability is guaranteed by the logical path associated with the STPA process. In other words, the approach will assure that a complete rationale is associated with each requirement in the form of ACCIDENT > HAZARD > SAFETY CONSTRAINT > DESIGN RECOMMENDATION > REQUIREMENT. The level of detail of the requirements can also be refined by applying STPA as an iterative process with an increasing level of detail at each iteration, potentially starting in the early concept development stage. This will ensure that a strictly top-down perspective is maintained during the entire requirements generation process aiming at capturing as many design issues as possible. This perspective is useful not only before the product is put in service, but also before the development gets to a stage where changes to current solutions are expensive or even infeasible and during operations and system evolution.

Control action L1 – Deactivate Air Conditioning Unit

This control action refers to a button generally located in the overhead panel of the cockpit that allows activating/de-activating the air conditioning units. Several unsafe control actions are associated with this control action. A specific example is analyzed here in order to show how the role of maintenance personnel can be included using STPA.

Unsafe control actions

The unsafe control action considered in this example is for control action L1 linked to hazard H2 (Table 5):

Table 5 – Unsafe Control Action for Control Action L1

ID	Control Action	Provided	No.	Not Provided	Too Late, Too Early, Wrong Order	Too long, too short
L1	Deactivate Air Conditioning Unit		211	The maintenance personnel do not deactivate the air conditioning unit before leaving the aircraft (e.g. before/after a shift) and closing the aircraft door		

The risk here is that the aircraft pressurizes on ground and it becomes impossible to re-open the door from outside. This is a real-case scenario that was identified with the participation of maintenance staff in the STPA analysis. Here is the related safety constraint:

SC211 The aircraft shall never be left with the air conditioning operating, no qualified personnel on board, and the aircraft door closed.

Causal Scenarios

Two primary causal scenarios have been identified for this unsafe control action (among others) (Table 6):

Table 6 – Causal Scenarios for UCA 211

U.C.A.	**211**
The maintenance personnel do not deactivate air conditioning unit before leaving the aircraft (e.g. before/after a shift) and closing the aircraft door (risk: aircraft pressurizes on ground and it is impossible to re-open the door).	
Scenarios	
1	During a hot or cold day, the maintenance team wants to perform their duty in a comfortable environment, therefore, before starting their shift, they activate air conditioning and leave the aircraft by closing the door.
2	During a hot or cold day, the maintenance team is performing their duty in a comfortable on-board environment. When the shift ends or the personnel needs to momentarily leave the aircraft, the aircraft door is closed after exiting.

The STPA analysis allows identifying this kind of off-nominal scenario by providing a framework in which all types of information can be included.

Design Recommendations and Requirements

The design recommendations and requirements to address this case are shown below (Table 7):

Table 7 – Design Requirements and Design Recommendations for UCA 211

Nbr		SAFETY CONSTRAINTS	
211		The aircraft shall never be left with the air conditioning operating, no personnel on board and aircraft door closed.	
Nbr		**DESIGN RECOMMENDATIONS**	
DR	211.1	Instruct maintenance personnel to never leave the aircraft door closed when air conditioning is active and no personnel are on-board	
DR	211.2	Design an external button to de-pressurize cabin from outside of the aircraft.	
Nbr		**REQUIREMENTS**	**TRACEABILITY TO DESIGN RECOMMENDATIONS**
R	211.1	A knob shall be installed in the non-pressurized electronic compartment of the aircraft to perform cabin dump on ground only.	DR211.2
R	211.2	The aircraft manual shall instruct not to allow the air conditioning to operate with no qualified maintenance personnel in the cockpit.	DR211.1
R	211.3	The maintenance personnel training shall instruct not to allow the air conditioning to operate with no qualified maintenance personnel in the cockpit.	DR211.1

Besides training and manual content, a feature could be added to the AMS system that allows depressurizing the aircraft from outside.

STPA in this case allows capturing information coming from the experience of maintenance personnel and address an issue that could affect operations.

Control action F6 – Deactivate Air Conditioning Unit

This control action is the same as the one previously analyzed, although it is associated with another controller: the pilot.

Unsafe control actions

The following unsafe control actions can be identified for control action F6 (Table 8):

Table 8 – Unsafe Control Actions for control action F6

ID	Control Action		Provided		Not Provided	Too Late, Too Early, Wrong Order		Too long, too short
F6	Deactivate Air Conditioning Unit	27	The pilot deactivates the air conditioning unit when a smoke source is present in the pressurized area	28	The pilot does not deactivate the air conditioning unit when this may affect minimum thrust availability (given altitudes, failure conditions etc.)		29	The pilot deactivates the air conditioning unit for too long when the external conditions cannot guarantee safety/comfort standards

	30	The pilot does not deactivate the air conditioning unit when cooling air is not provided to the heat exchanger
	31	The pilot does not deactivate the air conditioning unit when leaving the aircraft and closing A/C door
	32	The pilot does not deactivate the air conditioning unit when dump is activated in case of smoke and insufficient outlet flow

These unsafe control actions are particularly interesting as they relate to a common problem in design and definition of safe operations: contexts highly influence the appropriateness of specific features of the system (in this case the possibility of de-activating the air conditioning units). Trade-offs are necessary and the controller (in this case the pilot) needs to be aware and "helped" by the system to understand when it is appropriate to perform a specific action or not. UCAs 27 and 32 are an example of what was just discussed: deactivating the air conditioning is generally necessary when smoke is present in the aircraft and dump has been activated to evacuate the smoke. However, if the altitude is too high to perform dumping and smoke is in the pressurized area of the aircraft, then air conditioning must be kept on in order to at least filter out some of the smoke. The pilot, however, may not be in the position to consider these factors during an emergency event such as smoke being detected on-board. The system should therefore be designed to help him or her make an appropriate decision.

Regardless of the final design strategy to support the pilot, note that STPA identifies the issue very early in the design process and therefore allows addressing it in a more strategic way than might be possible after the design is more complete or waiting for operations to start. UCA 28 relates to a specific danger of the air conditioning units limiting the thrust available in specific conditions, while UCA 29 refers to a situation in which the crew has lost awareness of what the current aircraft temperature is (on ground or in-flight). Both these scenarios could be addressed by the design introducing specific alert messages to warn the pilot of possible problem. UCA 31 is the parallel of the UCA 210, but coming from the pilots and not the maintenance team.

In the rest of the section, UCA 30 is analyzed in detail. The unsafe context identified here is the following: the cooling circuit of the heat exchanger of the air conditioning unit could become obstructed. This can result in a surge of the fan present in the unit. The surge of the fan leads the compressor and turbine mounted on the same shaft to fret against the walls of the turbomachine and thus generate some smoke, which can be carried all the way to the cabin environment. The safety constraint (linked to hazard H1, H3, H4) is in this case:

SC30 The air conditioning units must be off when the heat exchanger is not being cooled off.

Causal Scenarios

The following causal scenarios are identified (Table 9):

Table 9 – Causal Scenarios for UCA 30

U.C.A.	**30**
The pilot does not deactivate the air conditioning unit when the cooling air is not provided to the heat exchanger	
Scenarios	
1	The pilot is not aware of the fact that no cooling air is provided to the heat exchanger because: 1) Feedback is missing / not been designed; 2) A feedback message exists, but is not clear; 3) A feedback message is masked by other messages.
2	The pilot knows no cooling air is provided to the heat exchanger, but he or she does not deactivate air conditioning because: 1) he/she is not aware of the impact this has on the air conditioning unit; 2) he/she does not know that deactivating the air conditioning would limit the damage.
3	The pilot knows no cooling air is provided to the heat exchanger and knows that deactivating the air conditioning would limit the damage, but he or she does not deactivate air conditioning because he or she estimates losing air conditioning at that altitude is too dangerous.
4	The pilot does not deactivate the air conditioning units because he or she detects smoke in the cabin and thinks the appropriate action to take is to keep the conditioning system on (UCA27).

Most of the scenarios identified relate to process model flaws in the controller. The pilot performs some actions because he or she is convinced f some conditions exist in the controlled process that instead are not true. The causes of these "false beliefs" are often related to missing or inadequate feedback (scenario 1), lack of specific knowledge about the system (scenario 2) and conflict with other decisions (scenarios 3, 4). It is important that the engineers take into consideration these complexities during design instead of shifting the whole responsibility onto the operator.

Design Recommendations and Requirements

The design recommendations for this UCA (Table 10) include the implementation of an alert message, of training content and of a physical sensor to provide the pilots with more specific information to update their model models.

Table 10 – Design Recommendations for UCA 30

No.	**SAFETY CONSTRAINTS**
30.1	The pilot must deactivate the air conditioning unit when cooling air is not provided to the heat exchanger
No.	**DESIGN RECOMMENDATIONS**
30.1	An alert should be implemented to inform the pilots of the loss of cooling air to the heat exchanger.
30.2	The alert should be associated with a procedure that asks the pilots to shut the air conditioning units off when possible and NOT perform an emergency descent. This aspect shall be emphasized during training as well.

30.3	Training should include information about when and how it is possible to operate without air conditioning. The procedure associated with the alert shall take this aspect into consideration as well.
30.4	Logic should be inserted in the AMS automation to shut down the air conditioning units when the heat exchanger is not being cooled adequately.
30.5	An accelerometer should be installed on the turbine/compressor to detect unit unbalance, so that a surge condition can be detected.

Some sample requirements are reported here below (Table 11):

Table 11 – Requirements for UCA 30

Nbr		SAFETY CONSTRAINTS	
30		The pilot must deactivate the air conditioning unit when cooling air is not provided to the heat exchanger	
Nbr		**DESIGN RECOMMENDATIONS**	
DR	30.1	An alert should be implemented to inform the pilots of the loss of cooling air to the heat exchanger.	
DR	30.2	The alert should be associated with a procedure that asks to shut the air conditioning units off when possible and NOT perform an emergency descent. This aspect shall be emphasized during training as well.	
DR	30.3	Training should include information about when and how it is possible to operate without air conditioning. The procedure associated with the alert shall take this aspect into consideration as well.	
DR	30.4	Logic should be inserted in the AMS to automatically shut down the air conditioning units when the heat exchanger is not being cooled enough.	
DR	30.5	An accelerometer should be installed on the turbine/compressor to detect unit unbalance, so that a surge condition can be detected.	
Nbr		**REQUIREMENTS**	**TRACEABILITY TO DESIGN RECOMMENDATIONS**
R	30.1	The AMS shall have a logic that detects loss of cooling of the heat exchanger.	DR30.4,DR30.1
R	30.2	When loss cooling of the heat exchanger is detected, an alert message shall be sent to the crew.	DR30.1
R	30.3	The alert should be accompanied by a procedure that asks the pilots to shut the air conditioning units off and NOT perform an emergency descent, unless required by other procedures.	DR30.2
R	30.4	The aircraft manual shall instruct the pilots to de-activate air conditioning units and not to perform an emergency descent when an alert of loss of cooling of the heat exchanger appears in the cockpit.	DR30.2
R	30.5	The AMS shall contain logic that commands air conditioning units to shut down when loss of cooling unit is detected.	DR30.4
R	30.6	An accelerometer shall be installed on the compressor of the air conditioning unit to detect vibrations induced by surge.	DR30.5

These requirements are high-level requirements and will need to be articulated into more detailed requirements that will characterize the final solution implemented.

Regardless of the specific recommendations made here and the associated requirements, it is important to note that STPA brings these issues to the attention of the engineers early in the development phase when it still easy and cheap to make decisions about the system design.

3 – Results

The requirements illustrated in the previous section are just a sample of a more complete application performed at Embraer on this generic AMS system. By analyzing each control action (52 in total) and completing the process until the design recommendation stage, the following results were obtained:

- About 200 Unsafe Control Actions with corresponding safety constraints;
- About 700 Design Recommendations and related requirements.
- Although all the issues identified by STPA had been captured by the Embraer process, with this application, some of them were found at lot earlier stages of the development (e.g. before testing, prototyping etc.).

Design recommendations can be used, as shown, to derive development requirements. The process was performed in a timeframe of a couple of months with the participation of a process facilitator (STPA facilitator) working full-time with other stakeholders of the engineering process:

- AMS system specialists (50% time): Supported the identification of the fundamental functionalities and characteristics of the AMS system, the writing of design recommendations and associated system requirements;
- System specialists of the interface systems (30% time): Provided information on integration issues between the AMS and their system of competence, facilitated identification of interface systems constraints (particularly from unsafe control actions that could have an impact outside of the AMS system);
- Human factors specialists (20% time): Provided support in making sure that human factors elements were fully taken into consideration during the analysis;
- Pilots (20% time): Provided information on operating issues concerning the AMS, especially current non-standard practices among pilots;
- Maintenance/production mechanics (10% time): Provided information on maintenance and production issues that are often overlooked during system design.

While these numbers may vary from organization to organization, they provide an estimate of the type of effort necessary to carry on the requirement generation methodology presented in this paper.

4 – Conclusions

This application confirmed in practice that STPA can be used as a requirement generating tool and that it possesses the following valuable characteristics:

- TOP-DOWN approach. The logical structure of STPA allows accomplishing one of the primary objectives in systems engineering: identifying the high-level goals of the system and cascading all its requirements from these goals. While the concept is simple in theory, it is often difficult to put in practice in a consistent way. STPA provides a well-defined process to accomplish this objective.STRUCTURED approach. Requirement generation in a systematic way can be challenging and is often achieved at a low level only (i.e. where analytic techniques exist to solve certain issues, such as structural calculations, aerodynamic calculations, flight control laws etc.). High-level as well as interface requirements are those particularly affected by this problem. The result is often that requirement sets are ill-defined [13]. STPA allows deriving requirements in a structured way in which a certain level of completeness and rigor in the analysis is guaranteed (see point below);
- TRACEABILITY of requirements. The proposed methodology also guarantees the complete traceability of any requirement by following the chain ACCIDENT > HAZARD > SAFETY CONSTRAINT > DESIGN RECOMMENDATION > REQUIREMENT. This is another major issue in systems engineering, that is addressed not only by documenting traceability “horizontally”, but also “vertically” i.e. at increasing levels of detail, as the methodology should be applied iteratively (e.g. after analyzing the interfaces of the AMS, the focus could be shifted to the AMS internal functional architecture).
- HUMAN-MACHINE INTERACTIONS analysis. Because of the flexibility provided by the controller and controlled process model, it is possible to analyze the role of humans in the system in conjunction with the physical component interactions and software. In other words, the human factors part of the analysis is not treated separately and with different techniques.
- COMPLETENESS of analysis. While it is impossible to guarantee full completeness of any analysis of a problem that cannot be solved deterministically (system functional architecture design), STPA allows identifying many

issues and capturing critical requirements very early in the development cycle of the aircraft. Testing and operations themselves are often the moment at which these emergent properties of the system manifest themselves and are identified as needing to be corrected. By using the methodology proposed in this paper, “early validation” of requirements can be performed based only on functional analysis. As evidence of the relative completeness of the STPA analysis, the list of requirements generated during this application was compared to existing Embraer documentation (based on ARP4754A process and traditional safety techniques contained in ARP4761). While the documentation was found to be complete, some of the requirements identified with STPA were only captured during more advanced development phases of the product.

- MULTIDISCIPLINARY approach. Because the methodology allows working at the same time on different system interfaces and it is expected that people from different areas of the engineering process will participate in the analysis, the kind of approach proposed is highly multidisciplinary. Additionally, the kind of conclusions drawn from the process can go beyond the safety aspect of the design and take into consideration other aspects of a product design (e.g. maintainability, durability, production constraints etc.).
- REUSE of analysis results. Keeping the level of abstraction high and limiting the analysis to functional aspects of the system allows defining a solid set of design recommendations that are almost architecture or solution independent and that can be reused in future developments. It could be possible to reuse the requirements too, depending on how architecture specific they are and how much of the new product will change with respect to the previous one. Given that traceability is also assured throughout the process, a sound rationale will be available for future applications as well.

Acknowledgements

The authors of this paper wish to express their gratitude to Embraer S.A. for sponsoring the project. Specifically, the authors wish to thank: Ricardo Moraes, Armando Carbonari, Marcos Antonio Viana Tavares, Ivanir Chappaz, Bruno Guedes Faria, Antonio Domiciano, Allan Ferreiros, Louise Novello Bätzner-Ribeiro, Gustavo Bertoli, Luis Henrique Santiago, and Ricardo Matsushima. The authors from MIT would also like to thank Embraer for allowing access to their facilities, documentation, and internal knowledge. This research was also partially supported by the MIT-Brazil association directed by Rosabelli Coelho-Keyssar.

References

[1] N. Leveson, Engineering a Safer World, Cambridge: MIT Press, 2012, pp. 7-60.

[2] ANAC, "Relatorio Final - A - N67/CENIPA/2009," July 2009. [Online]. Available: http://www2.anac.gov.br/arquivos/RF3054.pdf.

[3] N. T. S. B. NTSB, "Descent Below Visual Glidepath and Impact With Seawall, Asiana Airlines Flight 214, Boeing 777-200ER, HL7742, San Francisco," NTSB, Washington, 2014.

[4] N. Leveson, Engineering a Safer World, Cambridge: MIT Press, 2012, pp. 14, 211-212.

[5] N. Leveson, Engineering a Safer World, Cambridge: MIT Press, 2012, pp. 211-249.

[6] N. Leveson, Engineering a Safer World, Cambridge: MIT Press, 2012, pp. 181-191.

[7] N. Leveson, Engineering a Safer World, Cambridge: MIT Press, 2012, p. 88.

[8] N. Leveson, "A Systems Approach to Risk Management Through Leading Safety," *MIT,* 2015.

[9] BAE, "Final Report On the accident on 1st June 2009 to the Airbus A330-203 registered F-GZCP operated by Air France flight AF 447 Rio de Janeiro - Paris," Bureau d’Enquêtes et d’Analyses pour la sécurité de l’aviation civile, Paris, 2012.

[10] DAC, "INFORME DE ACCIDENTE DE LA AERONAVE AIRBUS A-340-600, OCURRIDO EN EL AEROPUERTO MARISCAL SUCRE DE QUITO, EL 9 DE NOVIEMBRE DE 2007," Dirección General de Aviación Civil, Quito, 2013.

[11] N. Leveson, Engineering a Safer World, Cambridge: MIT Press, 2012, p. 217.

[12] J. Scavini, "Airbus A320 family, Wikipedia," Wikipedia, 15 06 2018. [Online]. Available: https://en.wikipedia.org/wiki/Airbus_A320_family. [Accessed 15 06 2018].

[13] J. D. &. G. M. Arthur, "An operational model for structuring the requirements generation process," *Requirements Engineering,* vol. 10, no. 1, pp. 45-62, January 2005.

[14] NASA, "Systems Engineering Handbook," Washington, D.C., 2007.