Architecting a Safety Case for UAS Flight Operations

Ewen Denney, Ph.D.; SGT / NASA Ames Research Center; Moffett Field, California, USA

Ganesh Pai, Ph.D.; SGT / NASA Ames Research Center; Moffett Field, California, USA

Keywords: Argument structures, Argument architecture, Bow-tie models, Safety architecture, Safety case, Unmanned aircraft systems

Abstract

Over the past few years, we have been developing safety cases for several NASA unmanned aircraft system (UAS) missions involving increasingly complex operational concepts. We have also begun including structured argumentation in the safety case reports to organize and explicitly document the reasons why the operations can be expected to be acceptably safe. Although each operation has particular mission-specific constraints and safety requirements, we have identified similarities amongst the associated hazard control mechanisms and safety arguments. The twin aims of this paper are to *a*) facilitate future reuse of the UAS operational safety measures and the associated safety arguments, and *b*) aid safety case comprehension and evaluation. Towards achieving these goals, we first present a generic concept for low altitude operations, describing the commonalities/differences between the missions, and the dependencies between the concrete details of specific missions and the applicable safety systems. Then we describe two architectural models: *i*) an abstract safety architecture specifying the collection of hazard controls, given using bow-tie diagrams, and *ii*) an argument architecture, given in terms of abstract argumentation patterns. We also discuss the relationship between the safety and argument architectures outlining their roles in creating the safety case and its underlying safety arguments.

Introduction

NASA's UAS Traffic Management (UTM) effort (ref. 1) is presently involved in developing air-traffic management technologies to enable small UAS (sUAS) to safely access and operate in low altitude uncontrolled airspace within the US National Airspace System (NAS). In brief, UTM is being engineered as a series of so-called *technical capability levels* (TCLs), each of which provides an increasing level of automation and autonomy to manage sUAS operations while seeking to maintain (or improve) the prevailing level of safety. The engineering plan calls for four TCLs at the end of each of which, a campaign of flight test demonstrations conducted within the NAS provides the proof of concept. TCL 1 was successfully demonstrated and concluded in Fall 2015, while the demonstration of TCL 2 is scheduled to occur in Fall 2016.

In general, these flight tests are subject to a number of requirements set forth by the Federal Aviation Administration (FAA), i.e., the US aviation safety regulator, as well as by NASA (though in this paper we are mainly concerned with the former). For instance, a *Certificate of waiver or Authorization* (COA), which is the authority to access and operate UAS in the NAS, is required. The FAA grants this authority to *public* entities, whereas for civil operations, the authority is a special airworthiness certificate, or an exemption from airworthiness, together with a COA (ref. 2). There are also a number of operational and regulatory requirements (ref. 3), e.g., the use of ground-based or airborne visual observers (VOs), maintaining direct two-way radio communications with the appropriate air traffic control (ATC) facilities, etc. Moreover, under certain conditions (e.g., utilizing an alternative means of compliance to the regulations, operating in certain controlled airspace classes, etc.) a system *safety case* is also required. The safety case expected is a type of safety risk management document that addresses, at a minimum (ref. 3): *a*) details about the system and environment, including existing procedures, operations, roles and responsibilities; *b*) the intended changes to the system, e.g., the introduction of new technology, equipment and procedures; *c*) UAS capabilities and airworthiness information, etc.; *d*) hazard and risk analyses (of the proposed changes) including details of the assumptions made, the criteria for categorizing hazards, the levels of initial and residual risk, hazard mitigations, risk treatment and hazard tracking; and *e*) details of safety risk management planning.

As such, due to the nature of the associated concept of operations (CONOPS)—which progressively relax the existing UAS operational constraints—COAs with supporting system safety cases are required to enable the forthcoming

UTM TCL flight test demonstrations. We have been directly involved in the development of those safety cases[1], leveraging our prior experience with the same in enabling beyond visual line-of-sight (BVLOS) UAS operations in Alaska (ref. 4). In addition to the required content, we have begun to include into the safety case, structured argumentation to explicitly document the rationale why the identified safety mitigations and requirements can be expected to reduce risk to an acceptable level. Based on these efforts, we have identified generic safety mechanisms that are common across the UAS operations (and the corresponding safety cases), although each mission has its own specific constraints and requirements. For the forthcoming missions (and corresponding safety cases), we want to be able to leverage and carefully reuse those safety assets to the extent possible.

Effectively, to achieve this reuse without disrupting the existing safety systems of the NAS, as well as to maintain the prevailing level of safety, a proper understanding is required of the architecture of the *overall* safety system that is pertinent to the sUAS operations being undertaken. In other words, there is a need to understand *i*) the dependencies amongst the constituent components of the overall safety system, *ii*) the relationships between the relevant (hazard) mitigation measures, and *iii*) how reuse and/or changes to those safety measures (including the introduction of new safety measures) will impact the overall system (and its safety case). Moreover, this must be communicated to the regulator to provide the assurance required that safety risks have been adequately managed. As the way forward, we want to provide a framework through which the required safety systems and the associated safety cases can be *architected*.

Motivating Context

First, we describe a generic concept for low altitude sUAS operations, which summarizes some of the commonalities and differences amongst the various UAS missions for which we have created safety cases. We also characterize the dependencies between the concrete details of specific missions and the applicable safety systems. Then, we present the key *safety concerns* applicable to this generic CONOPS along with a number of risk mitigation *barriers*.

Generic Operational Concept: We characterize the operational concept for low altitude sUAS operations in terms of the operating environment (i.e., the airspace), the mission characteristics, and the aircraft involved.

Airspace of Operations: In general, low altitude operations have typically involved the airspace from the surface to approximately 2500 feet above ground level (AGL), within a single *operating range* (OR)—a three dimensional, polygonal airspace volume. Depending on the location of the OR, the airspace may be *controlled* (i.e., the airspace classes A through E), *uncontrolled* (i.e., Class G airspace), or a combination of the two. For instance, low altitude airspace in the proximity of a non-towered airport with a published instrument approach procedure will usually straddle both the controlled Class E, and the uncontrolled Class G airspaces.

From a safety standpoint, controlled airspace affords additional safety mechanisms, including separation services provided by air traffic control (ATC) to those aircraft meeting the requirements for flying in controlled airspace. In contrast, aircraft in uncontrolled airspace often may not be visible to ATC, usually observe visual flight rules (VFR) and, therefore, a key component of safety is too *see and avoid* other aircraft that may be in conflict. Indeed, one of the current operational constraints for low altitude sUAS operations, is the use of (ground-based or airborne) visual observers as the means of compliance with the 'see and avoid requirement' of the federal aviation regulation (FAR) 14 CFR 91.113. In fact, operations that relax this requirement must submit a safety case as part of the FAA operational approval process. The variation in the OR size, shape, and elevation also poses safety implications, when requirements are to be defined, and solutions implemented, for both airspace surveillance and airspace conflict resolution/avoidance.

Takeoff/landing locations may or may not be located within the OR, and a transit corridor may be defined to facilitate flight between a separate launch/landing area and the OR. The OR and transit corridor may be each located over land, water, or both. Additionally, it may be close to an airport, and enclosing some population, structures, and road traffic. Missions involving operations in the vicinity of an active airfield (without a control tower) impose specific communication and coordination requirements that differ from those that apply when operating farther away from the airfield. Thus far, none of the operations have yet involved urban locations with a high population density and built-up areas, major airports, or airports with operating control towers. Although most of the missions have oc-

[1] Some of which are currently under review with the FAA.

curred within the NAS, some of them have involved transit through the NAS to/from international airspace, where operations were conducted under so-called due regard rules (refs. 4, 5). This imposes additional requirements (again, on communication and coordination as a safety mitigation strategy for airspace deconfliction) due to flight occurring through the US *air defense identification zone* (ADIZ) during return transit.

Mission Characteristics and Aircraft: Operations have consisted of flights with fixed-wing or rotary-wing unmanned aircraft (UAs) that weigh ≤ 55 pounds and that have a maximum airspeed of 100 knots. There may be multiple aircraft operating concurrently in a defined sub-volume of the OR, either within visual line of sight (VLOS), or beyond VLOS (BVLOS). Additionally, flight plans may include one-way or returning flights, with single or multiple point-to-point segments. The former characterizes a flight profile involving a single takeoff location followed by a flight to a defined landing location (which can also be the takeoff location), while the latter entails multiple takeoff and landing locations within a single flight plan. The class of operations considered have been restricted to daytime operations under visual meteorological conditions (VMC) suitable for operating under the (stricter) visual flight rules (VFR) of Class E airspace.

The performance characteristics of the UAs, and more generally their airworthiness, constrain the operations and the associated safety requirements. For instance, a lower-level of airworthiness necessitates the definition of an OR that is sparsely populated (or unpopulated), and that does not contain (m)any built-up areas. Additionally, it also limits the avoidance maneuvers/procedures that can be defined.

Safety Concerns and Mitigation Barriers: Based on this generic CONOPS, a number of potential *safety concerns*[2] can be identified, including *a*) loss of safe separation between a non-cooperative aircraft and a UA, between UAs, and between UAs and terrain/terrestrial entities, *b*) UA flyaway/exiting the OR, *c*) inclement weather, and *d*) global navigation satellite system (GNSS) signal unavailability. In addition, a number of contributory safety concerns also exist concerning the UAS and its constituent systems and functions, e.g., loss of navigation capabilities, loss of the command and control (C2) link, airborne system failures and unexpected behaviors, hostile takeover, etc. As the concept is crystallized for a specific operating airspace, class of aircraft, and mission, some of these cease to be concerns. For instance, loss of safe separation between UAs, or between UAs and terrain/terrestrial entities is only a safety concern when operations occur over populated areas where there is a risk of harm to the population and/or damage to structures and property. We note that this is not a comprehensive list, but we believe they are broadly representative of the safety issues that can arise for the identified concept.

Several mitigation barriers can be employed to reduce the associated safety risk. Again, this list is not comprehensive, and is mainly indicative of the barriers that we have found applicable and have utilized in practice. These include: *i*) choice of the OR, *ii*) airworthiness, flight readiness and crew qualification, *iii*) special equipage for the UAS platforms, *iv*) communication and coordination, *v*) separation standards, *vi*) surveillance, *vii*) conflict resolution/deconfliction, *viii*) redundancy, and *ix*) standard, and off-nominal, operating procedures. Previously (ref. 7), we have adopted other safety barriers, in addition to those listed above. In general, much like the safety concerns, the applicability of the barriers depends upon the concretion of the generic concept. Moreover, once the barriers have been identified, we can add to the list of safety concerns considering the ways in which the applicable barriers can be compromised. Next, we describe how these barriers can be used to manage the identified safety concerns.

Approach

In creating the safety cases for the different low altitude sUAS operations, we have mainly undertaken the traditional aviation safety risk management activities (ref. 6) of hazard identification, risk analysis and assessment, and risk control as required. In particular, we conducted a preliminary hazard analysis (PHA). However, we have also been using two types of models/diagrams to augment this activity; namely, *a*) bow-tie diagrams, and *b*) argument structures and patterns. The purpose of the former is to model the relationship between the identified safety concerns and the applicable safety barriers so as to obtain a high-level overview of the overall safety system[3]. The latter serves to capture the underlying safety rationale. Here, we have been aided by our earlier work in developing UAS assurance arguments (ref. 8).

[2] As per traditional safety analysis terminology in aviation (ref. 6) these are, in fact, *hazards*. However, for this paper we adopt a slightly different, but compatible, notion of hazard based on bow-tie diagram terminology, clarified later in the paper.

[3] Although, our usage has been largely ad hoc as we gather more experience with bow-tie modeling.

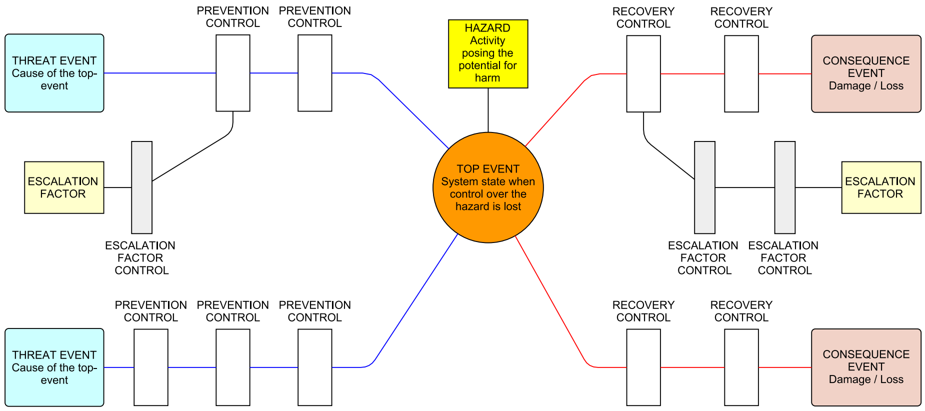
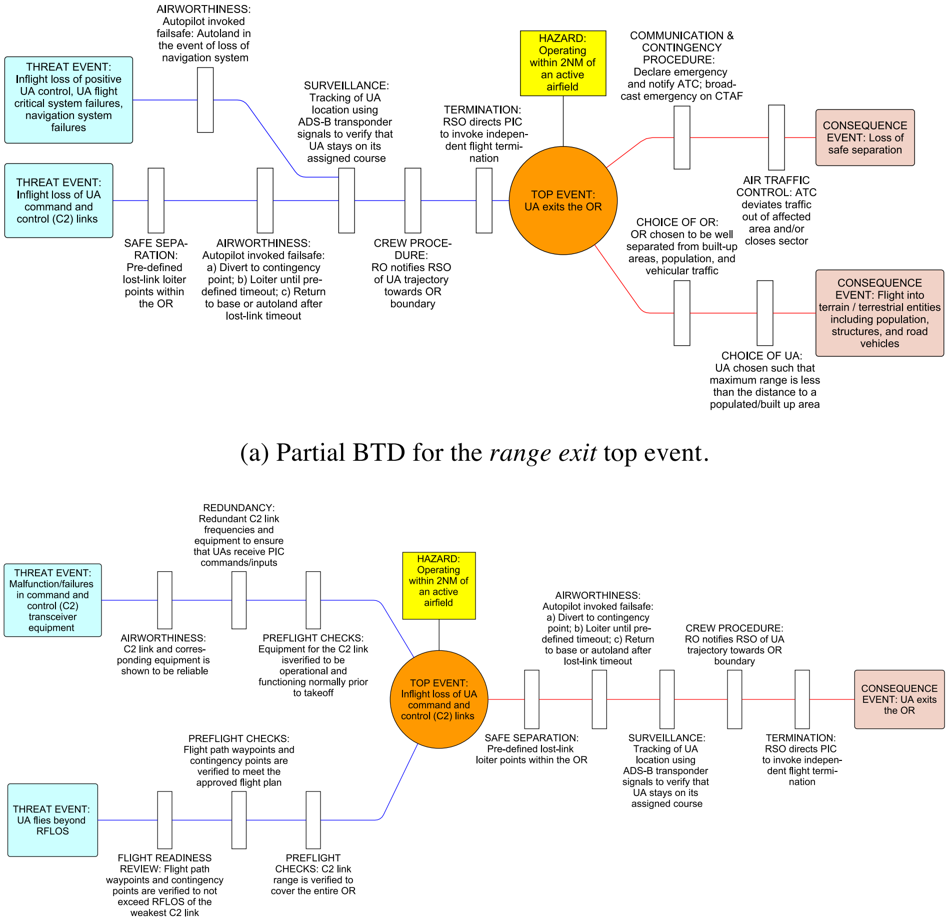

PREVENTION CONTROL
PREVENTION CONTROL
HAZARD Activity posing the potential for harm
RECOVERY CONTROL
RECOVERY CONTROL
THREAT EVENT Cause of the top-event
CONSEQUENCE EVENT Damage / Loss
ESCALATION FACTOR
TOP EVENT System state when control over the hazard is lost
ESCALATION FACTOR
ESCALATION FACTOR CONTROL
ESCALATION FACTOR CONTROL
ESCALATION FACTOR CONTROL
PREVENTION CONTROL
PREVENTION CONTROL
PREVENTION CONTROL
RECOVERY CONTROL
RECOVERY CONTROL
THREAT EVENT Cause of the top-event
CONSEQUENCE EVENT Damage / Loss

Figure 1 — Example bow-tie diagram (BTD) showing a hazard, its top event, threats, consequences, prevention and recovery (primary) controls, their escalation factors and the corresponding escalation factor (secondary) controls.


AIRWORTHINESS: Autopilot invoked failsafe: Autoland in the event of loss of navigation system
THREAT EVENT: Inflight loss of positive UA control, UA flight critical system failures, navigation system failures
SURVEILLANCE: Tracking of UA location using ADS-B transponder signals to verify that UA stays on its assigned course
TERMINATION: RSO directs PIC to invoke independent flight termination
HAZARD: Operating within 2NM of an active airfield
COMMUNICATION & CONTINGENCY PROCEDURE: Declare emergency and notify ATC; broadcast emergency on CTAF
CONSEQUENCE EVENT: Loss of safe separation
THREAT EVENT: Inflight loss of UA command and control (C2) links
TOP EVENT: UA exits the OR
AIR TRAFFIC CONTROL: ATC deviates traffic out of affected area and/or closes sector
CHOICE OF OR: OR chosen to be well separated from built-up areas, population, and vehicular traffic
SAFE SEPARATION: Pre-defined lost-link loiter points within the OR
AIRWORTHINESS: Autopilot invoked failsafe: a) Divert to contingency point; b) Loiter until predefined timeout; c) Return to base or autoland after lost-link timeout
CREW PROCEDURE: RO notifies RSO of UA trajectory towards OR boundary
CONSEQUENCE EVENT: Flight into terrain / terrestrial entities including population, structures, and road vehicles
CHOICE OF UA: UA chosen such that maximum range is less than the distance to a populated/built up area
REDUNDANCY: Redundant C2 link frequencies and equipment to ensure that UAs receive PIC commands/inputs
THREAT EVENT: Malfunction/failures in command and control (C2) transceiver equipment
HAZARD: Operating within 2NM of an active airfield
AIRWORTHINESS: C2 link and corresponding equipment is shown to be reliable
PREFLIGHT CHECKS: Equipment for the C2 link isverified to be operational and functioning normally prior to takeoff
AIRWORTHINESS: Autopilot invoked failsafe: a) Divert to contingency point; b) Loiter until predefined timeout; c) Return to base or autoland after lost-link timeout
CREW PROCEDURE: RO notifies RSO of UA trajectory towards OR boundary
TOP EVENT: Inflight loss of UA command and control (C2) links
CONSEQUENCE EVENT: UA exits the OR
SAFE SEPARATION: Pre-defined lost-link loiter points within the OR
SURVEILLANCE: Tracking of UA location using ADS-B transponder signals to verify that UA stays on its assigned course
TERMINATION: RSO directs PIC to invoke independent flight termination
PREFLIGHT CHECKS: Flight path waypoints and contingency points are verified to meet the approved flight plan
THREAT EVENT: UA flies beyond RFLOS
FLIGHT READINESS REVIEW: Flight path waypoints and contingency points are verified to not exceed RFLOS of the weakest C2 link
PREFLIGHT CHECKS: C2 link range is verified to cover the entire OR

(a) Partial BTD for the *range exit* top event.

(b) Partial BTD for the *loss of the command and control (C2) links* top event.

Figure 2 — Fragments of BTDs for two top events associated with the hazardous activity of operating within 2 nautical miles (NM) of an active airfield. Note that not all threat and consequence events have been shown, nor have any escalation factors and their corresponding controls.

Modeling Safety Risk Management: *Bow-tie diagrams* (BTDs), or so-called *barrier bow-tie* models, are a visual approach to modeling how safety risks can be managed. They have been applied in civil aviation for operational risk assessment and management (refs. 9, 10). They have also been used to structure generic safety cases addressing the mitigation of UAS operational risks, e.g., when supporting natural disaster response with UAS (ref. 11), and mid-air collision risk in civil airspace (ref. 12). BTDs offer seven key constructs (as shown in Figure 1), which we describe next. Figure 2 shows two concrete example BTDs for one of the NASA UAS operations occurring in the proximity of an active (non-towered) airfield. For brevity, the examples of Figure 2 do not comprehensively characterize the safety situations being modeled.

Hazard: This is an activity, condition, or entity that reflects a normal or desirable aspect of the concept of operations, but potentially can be a source of harm if control is lost. In Figure 2a, the hazard is operating within 2 NM of an active airfield, which is inherently hazardous when operational control is lost. Such a notion of hazard is subtly different from the *traditional* notion, e.g., as given in (ref. 6). The latter is usually given as "any real or potential condition that can cause harm" and reflects a *top event* (described next) in bow-tie terminology.

Top Event: This is the system state at which control over the hazard is lost, and there can be one or more top events for any given hazard. For example, in Figure 2a the top event for the given hazard is the UA exiting the OR. Note that the top events can be mapped to the *safety concerns* identified earlier.

Threats or Threat Events: These are the causes/sources of the top event that lead to its occurrence. For instance, in Figure 2a, a potential threat event for the given top event is a loss of the command and control (C2) link between the ground control station (GCS) and the airborne UA.

Consequences or Consequence Events: These are the potential outcomes resulting from the occurrence of the top event that, in turn, result in loss, damage, and/or harm. In Figure 2a, two consequences have been indicated, namely a loss of safe separation, and flight into terrain.

Controls: These are the mitigation measures taken *i*) to eliminate the threat events, or prevent threats from manifesting into top events, or *ii*) to manage the top event (once it has occurred) and prevent it from progressing into the consequence event states. The former are proactive, *prevention* controls, while the latter are reactive, *recovery* controls. A combination of controls of the same type/kind can grouped (abstractly) into *barriers*. For example, a surveillance barrier can utilize various sensors, such as radar and visual observers, each fulfilling a specific role for detecting airborne threats. We will consider threat/consequence event controls to be primary controls. In much the same way that controls can be grouped into barriers, barriers may be grouped into barrier *categories* where the category reflects a barrier allocation to some component of the overall system.

Escalation Factors: These are the weaknesses/vulnerabilities in the identified controls that can result in the control being ineffective or, more generally, a barrier being breached. Escalation factors are analogous to threat events and can represent, for example, failure modes for a specific control/barrier. For example, failure of the data link between the radar unit and its display is an escalation factor that can render ineffective, electronic surveillance of the airspace.

Escalation Factor Controls: These are secondary controls that manage escalation factors to arrest their progression into barrier breaches.

As we will see later in the paper, the collection of BTDs intuitively provides an abstract, high-level *architecture* of the overall safety system, in terms of the hazard controls, their organization, and the safety concerns being managed. Next, we describe how we explicitly capture the rationale underlying the safety system, i.e., the reasons why the particular collection and combinations of the safety barriers can be expected to enable safe operations.

Capturing Safety Rationale: We use *structured arguments* to document safety rationale, and *argument patterns* to represent abstractions of such reasoning. Thus, our vision of a safety case includes a structured and evolving argument that comprises explicit safety claims, assimilates heterogeneous safety-substantiating evidence, and presents the reasoning required to conclude that a system will be safe for a defined application and operating environment.

Structured Arguments: An *argument* is a connected series of propositions used in support of the truth of an overall proposition. We refer to the latter as a *claim*, whereas the former represents a *chain of reasoning* connecting the claim and the *evidence*. We present an argument structure as a diagram using the *goal structuring notation* (GSN) (ref. 13). Figure 3 shows an argument fragment as a directed acyclic graph of different GSN nodes and links. The different node types are: *goals* (shown as rectangles), *strategies* (shown as parallelograms), *contexts* (shown as rounded rectangles), *assumptions*, *justifications* (not shown in Figure 3, they are visually rendered as ellipses), and *solutions* (shown as circles). These six node types comprise the *core* constructs to capture the elements of an argument. In general, nodes refer to external items including:

- artifacts such as hazard logs, requirements documents, design documents, various relevant models of the system, etc.
- the results of engineering activities, e.g., safety, system, and software analyses, various inspections, reviews, simulations, and verification activities including different kinds of system, subsystem, and component-level testing, formal verification, etc., and
- records from ongoing operations, as well as prior operations, if applicable.

Links specify *support* (—▶) or *contextual* (—▷) types of relationships between the nodes. GSN also has other abstractions and notational extensions for modularity (ref. 13), though we will not cover those here.

The argument in Figure 3 concerns the rationale why a specific system event—i.e., an intruder descending into the radar cone of silence at a high altitude—can be ruled out as a credible threat (goal node G57). The approach (strategy node S19) is to show that the descent rate required to descend into the threat volume (TV)—i.e., the volume of airspace where an intruder is classified as posing a credible threat to the UA—is also not credible. As a result, it has now to be shown (Goal G47) that the required descent rate is much greater than the typical safe descent rate. This assertion requires the TV altitude, the traffic pattern altitude, and the typical descent rate at pattern altitude as context (context nodes C42, C44 and C45 respectively). The corresponding claim is shown by computing the required descent rates and by comparing that computed value with the typical descent rate (strategy node S20).

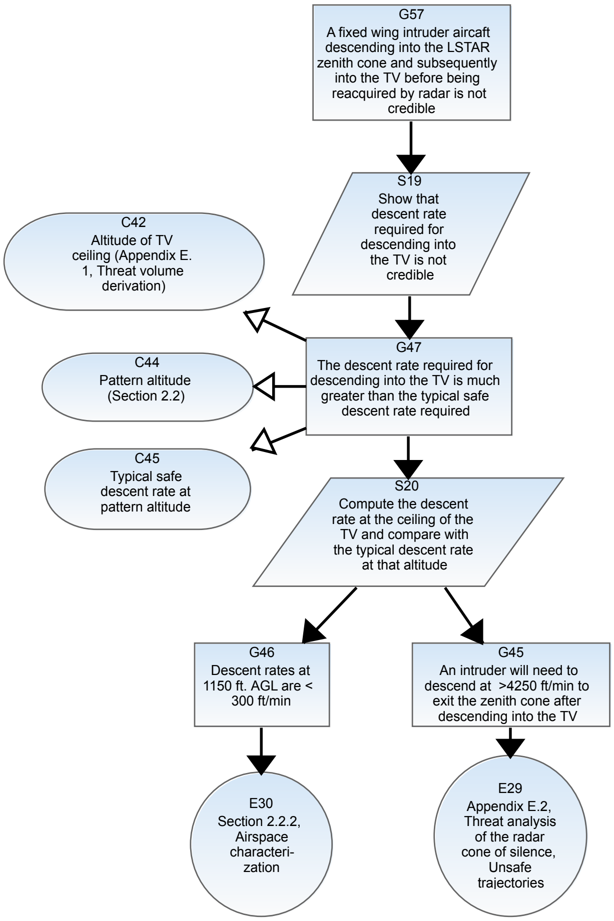

G57
A fixed wing intruder aircaft descending into the LSTAR zenith cone and subsequently into the TV before being reacquired by radar is not credible
S19
Show that descent rate required for descending into the TV is not credible
C42
Altitude of TV ceiling (Appendix E. 1, Threat volume derivation)
G47
The descent rate required for descending into the TV is much greater than the typical safe descent rate required
C44
Pattern altitude (Section 2.2)
C45
Typical safe descent rate at pattern altitude
S20
Compute the descent rate at the ceiling of the TV and compare with the typical descent rate at that altitude
G46
Descent rates at 1150 ft. AGL are < 300 ft/min
G45
An intruder will need to descend at >4250 ft/min to exit the zenith cone after descending into the TV
E30
Section 2.2.2, Airspace characteri-zation
E29
Appendix E.2, Threat analysis of the radar cone of silence, Unsafe trajectories

Figure 3 — Fragment of a GSN argument structure (from the overall argument for why a surveillance barrier meets its safety requirements), describing the rationale why a specific airspace situation can be eliminated as a credible threat.

The solution nodes (solution nodes E29 and E30) and the corresponding evidence assertions (goal nodes G45 and G46, respectively), provide the necessary evidence for the top-level claim—that the required descent rate would need to be at least an order of magnitude greater than the safe descent rate: an extremely unlikely event, and therefore not a credible safety concern. Note that this argument fragment is part of a larger argument, concerning the assertion that a surveillance barrier (used in the same UAS operations pertinent to the bow-tie diagrams of Figure 2) meets its safety requirements. A zoomed out view of this larger argument has been shown as the GSN structure in Figure 8, where the nodes highlighted with the thick dark edges correspond to the argument fragment of Figure 3.

Argumentation Patterns: Argumentation patterns are intended to capture repeatedly used structures of successful arguments, providing a reusable approach to safety argumentation. That is, they provide a way to capture expertise, known best practices, successful certification approaches, and solutions that have evolved over time. In brief, we can also represent argumentation patterns using GSN—and, visually, they look similar to GSN arguments—although additional node annotations, and link types are available, along with a means to express the notions of parameterization, multiplicity, choice, and iteration, which are required for abstraction. Although we will rely upon argumentation patterns to create an *argument architecture* (described later in this paper), the specific details of their syntax, semantics, construction, and use—which we have given in more detail elsewhere (ref. 14)—are not in scope for this paper.

Architectural Models

Safety Architecture: We now introduce a notion of an *abstract safety architecture* with the aim of characterizing the overall combination of mechanisms available for safety risk management of the generic CONOPS for low altitude sUAS operations. Our notion centers on using BTDs, building upon them by aggregating multiple diagrams, and also providing various views through those diagrams, so as to present an overarching perspective of the entire safety system.

Preliminaries: There are a number of nuances about the semantics of BTDs. For instance, the visual depiction of controls occurring in sequence need not imply that the controls are sequential. Rather, there can be parallel (i.e., alternative) controls where only one of the specified controls is used at a given time. Hazard controls, and consequently barriers, need not be independent. For example, controls comprising avoidance maneuvers will not be effective if controls in the surveillance barrier are affected. BTDs usually do not visually distinguish between parallel and sequential controls, nor do they depict barrier dependencies.

As can be seen from Figure 1, a BTD can be considered as showing a snapshot of a chain of events. Indeed, by moving the focus earlier in an event-chain, a top event can be seen as a consequence event, whereas a threat event can be seen as the top event leading to that new consequence. Likewise, the causes of the erstwhile threat event are, now, the new threat events. The same applies for controls, escalation factors, and escalation factor controls (with respect to their focus on prevention/recovery). This lends itself to a notion of linking or *chaining* BTDs.

Recall that a hazard (i.e., a hazardous activity) can be associated with multiple top events. Similarly, a threat event can lead to multiple top events, while a top event can have multiple consequences. The various relationships between the bow-tie elements gives a framework to specify an abstract safety architecture that describes the organization of the safety system(s) in terms of the controls, barriers and barrier categories that they (will) implement, and the safety concerns that they (will) address. Thus,

- In our approach, a safety architecture consists of a collection of interconnected BTDs, each of which has a unique top event associated with a given hazard.
- We omit a formal definition of safety architecture here and simply assume that each BTD comprises a set of events, each of which has type *threat*, *top*, or *consequence*, and an ordering '<', such that for the BTD there is a unique top event, all threat events precede that top event, and all consequence events succeed the top event. Additional structural properties (omitted here) relate events, controls, and their escalation factors.
- Those BTDs which are associated with a common hazard can share events (both threat and consequence), controls, as well as escalation factors and their controls. Moreover, BTDs for different hazards should *not* share any events, although they can share controls. In practice, these rules may be relaxed (e.g., BTDs for separate hazards might share elements), but ideally this is not the case.

Figure 4 shows how different BTDs can be related. These BTDs are, in fact, the same as those in Figure 2, though here we only give the content of the hazard and the related events, and omit the descriptions of the hazard controls for illustrative purposes. It can be seen that the two diagrams share hazards, and moreover they share events and controls (shown as the dashed box in Figure 4). Specifically, the top event node of the BTD addressing the OR exit top event, is a consequence of the second BTD (whose top event concerns inflight loss of the C2 links). Likewise, a threat event of the former, is the top event of the latter. Also, the recovery controls of the latter are the prevention controls of the former. In general, given a set of safety issues for an operational concept, it is intuitive to see that the BTDs corresponding to the top events of interest can be related and have events and controls in common. Moreover, the combination of these summarizes the various dependencies between all the controls in the overall safety system. There are additional *compatibility* conditions relating those BTDs that share elements. We will describe these next, when we discuss diagram *composition*.

Composition of BTDs: As noted above, diagrams for the same hazard can share events and controls. We now describe a way of *composing* these diagrams to give a single diagram. This mechanism thus provides a means of constructing complex diagrams from simpler ones, or "collapsing" the architecture. Figure 5 shows a composition where a top event in one diagram is mapped to a threat event in another, and a top event in one is mapped to a consequence in the other. We can also compose by mapping a threat event in one to a consequence in another. In general, we can *chain* diagrams together in this manner so long as they satisfy certain minimal compatibility conditions.

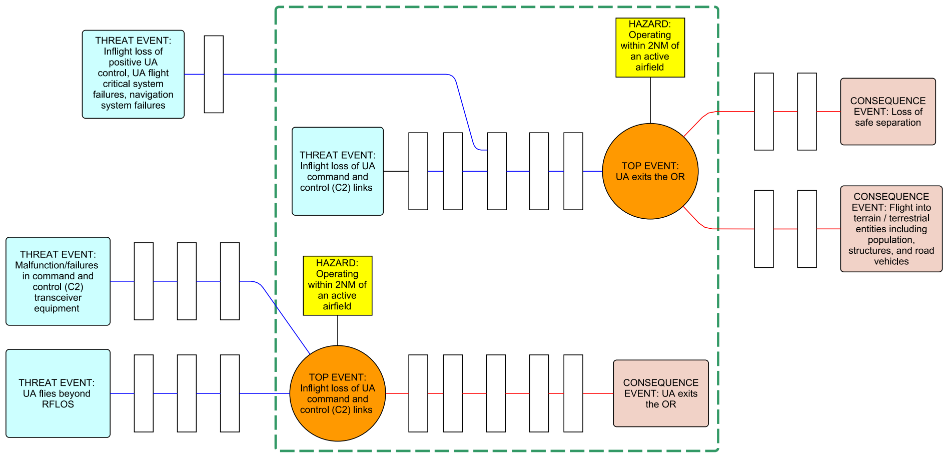

THREAT EVENT: Inflight loss of positive UA control, UA flight critical system failures, navigation system failures
HAZARD: Operating within 2NM of an active airfield
THREAT EVENT: Inflight loss of UA command and control (C2) links
TOP EVENT: UA exits the OR
CONSEQUENCE EVENT: Loss of safe separation
CONSEQUENCE EVENT: Flight into terrain / terrestrial entities including population, structures, and road vehicles
THREAT EVENT: Malfunction/failures in command and control (C2) transceiver equipment
HAZARD: Operating within 2NM of an active airfield
THREAT EVENT: UA flies beyond RFLOS
TOP EVENT: Inflight loss of UA command and control (C2) links
CONSEQUENCE EVENT: UA exits the OR

Figure 4 — Bow-tie diagrams of Figures 2a and 2b (with only the events shown unabridged). The dashed box indicates elements common to both diagrams.

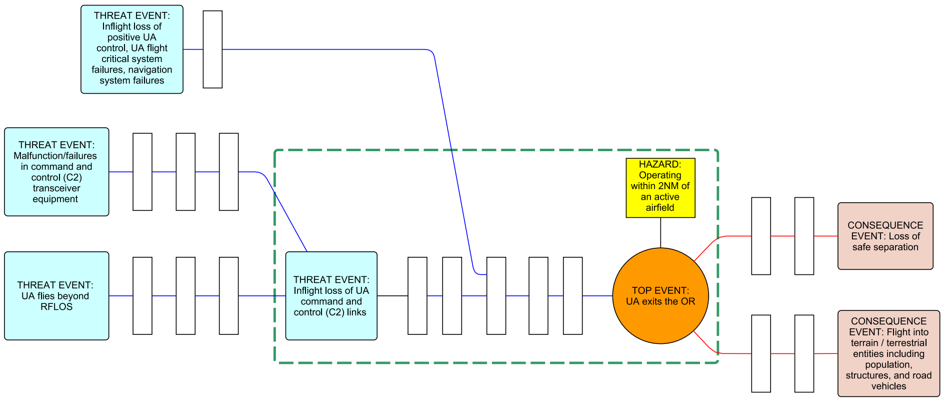

THREAT EVENT: Inflight loss of positive UA control, UA flight critical system failures, navigation system failures
THREAT EVENT: Malfunction/failures in command and control (C2) transceiver equipment
HAZARD: Operating within 2NM of an active airfield
CONSEQUENCE EVENT: Loss of safe separation
THREAT EVENT: UA flies beyond RFLOS
THREAT EVENT: Inflight loss of UA command and control (C2) links
TOP EVENT: UA exits the OR
CONSEQUENCE EVENT: Flight into terrain / terrestrial entities including population, structures, and road vehicles

Figure 5 — Composition of bow-tie diagrams, with the dotted box indicating where the original participants in the composition (Figures 2a and 2b) were combined.

We say that two BTDs, A and B, are *compatible* whenever if $e_1 <_A e_2$ and $e_1, e_2 \in B$ then $e_1 <_B e_2$, and if $e_1 <_B e_2$ and $e_1, e_2 \in$ A then $e_1 <_A e_2$ where $<$ is the union of the orderings $<_A$ and $<_B$. That is, if two events are related in one diagram, and appear in the other, then they must be related in the other, and vice versa. This prevents incompatible orderings (and also that the events be related in one, but in separate branches in the other). It does allow the events to be directly linked in one diagram but to have intermediate events in the other. Similarly, though we allow different events to share controls, we need consistent ordering of controls relative to events, that is, if $e < c$ then $c \not< e$. We do not impose any compatibility conditions on escalation factors, though escalation factor controls and (primary) controls should be disjoint. Ideally, controls would not share escalation factors, i.e., there would be no common cause failures, but in general we cannot assume this.

Then, given two compatible BTDs, we compose them simply by merging the events and controls and choosing a new top event to be any node e_{top} such that for all e, $e \leq e_{\text{top}}$ or $e \geq e_{\text{top}}$. Syntactically, this top event can be anywhere that is valid though, in practice, it is most likely to be between the top events of the component diagrams. This implies that the result cannot be such that a consequence event is to the left of a top event, or a threat event is to the right of a top event. If no such event exists, then the diagrams can not be composed. We then reassign the types of the various events so that all events that precede e_{top} are threats, and all that succeed it are consequences. A different kind of composition is where the top event of one diagram corresponds to the failure of a control in another, and threat events in the former correspond to escalation factors of the control in the latter.

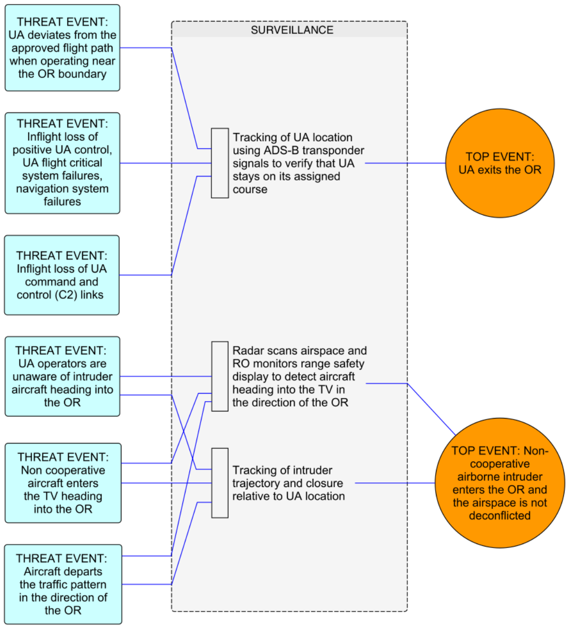

THREAT EVENT: UA deviates from the approved flight path when operating near the OR boundary
THREAT EVENT: Inflight loss of positive UA control, UA flight critical system failures, navigation system failures
THREAT EVENT: Inflight loss of UA command and control (C2) links
THREAT EVENT: UA operators are unaware of intruder aircraft heading into the OR
THREAT EVENT: Non cooperative aircraft enters the TV heading into the OR
THREAT EVENT: Aircraft departs the traffic pattern in the direction of the OR
SURVEILLANCE
Tracking of UA location using ADS-B transponder signals to verify that UA stays on its assigned course
Radar scans airspace and RO monitors range safety display to detect aircraft heading into the TV in the direction of the OR
Tracking of intruder trajectory and closure relative to UA location
TOP EVENT: UA exits the OR
TOP EVENT: Non-cooperative airborne intruder enters the OR and the airspace is not deconflicted

Figure 6 — Example of a *view d*erived from the safety architecture.

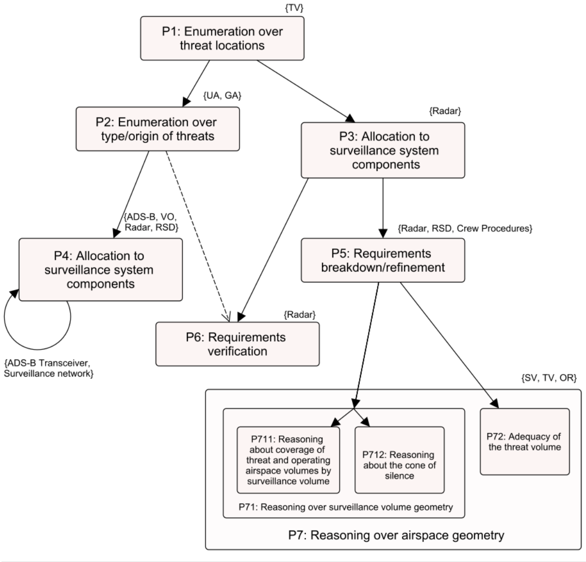

{TV}
P1: Enumeration over threat locations
{UA, GA}
P2: Enumeration over type/origin of threats
{Radar}
P3: Allocation to surveillance system components
{ADS-B, VO, Radar, RSD}
P4: Allocation to surveillance system components
{ADS-B Transceiver, Surveillance network}
{Radar, RSD, Crew Procedures}
P5: Requirements breakdown/refinement
{Radar}
P6: Requirements verification
{SV, TV, OR}
P711: Reasoning about coverage of threat and operating airspace volumes by surveillance volume
P712: Reasoning about the cone of silence
P71: Reasoning over surveillance volume geometry
P72: Adequacy of the threat volume
P7: Reasoning over airspace geometry

Figure 7 — Example representation of an argument architecture.

In this case we can map the chains from threat events to top events into the latter diagram. If the top event in the latter appears in a consequence leg of the former then this can be added as an additional consequence leg in the latter, otherwise the consequence leg is discarded in the composition.

Views: A number of *views* can be derived from the abstract safety architecture to emphasize specific perspectives, contributing to the design of the safety system. One possible view (for example, as shown in Figure 6) is a *slice* across the safety architecture, which gathers all the threat events being managed by a specific control and/or barrier, and also lists the top events that may result if the barrier is breached. This view can be thought of as indicating a specification of the barrier functionality at a system level. This view conveniently presents all the safety concerns being addressed for a specific barrier, and can be useful in communicating to the regulator what a new safety system is intending to address. Analogous to this is a view where all the top events and consequence events related to a specific control/barrier are presented. Such a view, we hypothesize, could also be useful in synthesizing standardized operating procedures for emergency situations, since it aggregates all procedural controls associated with the procedural mitigation barrier. A third view presents all the top events resulting from a single threat event. This is effectively the event chain beginning from a single threat event, across the entire system. We believe that such a view can be useful to focus the safety discussion on specific high priority threat events presenting all the safety assets available and how they are organized to manage that threat. A related view shows all the top events leading to a particular consequence/chain of consequences. Beside these, other views can also be defined focusing on different elements (or combination of elements) of the safety architecture, including the escalation factors and their controls.

Argument Architecture: We now discuss the architecture of the rationale underlying why the safety architecture, i.e., the collection and combination of barriers, is expected to reduce risk. The *argument architecture* refers to the high level organization of the overall safety rationale, characterizing the reasoning and intent of the various components of the argument. Here we use argumentation patterns (described earlier) to specify the same, although others have also proposed the use of modular arguments (ref. 15) for that purpose.

Figure 7 shows the argument architecture representing the structure of the assurance argument for the surveillance barrier used for the NASA UAS operations occurring in the proximity of an active (non-towered) airfield (i.e., the same operations whose BTDs are shown in Figure 2). From the figure, we see that the argument architecture consists of a hierarchical directed graph, where each node represents a fragment of reasoning that can be characterized by an argument pattern.

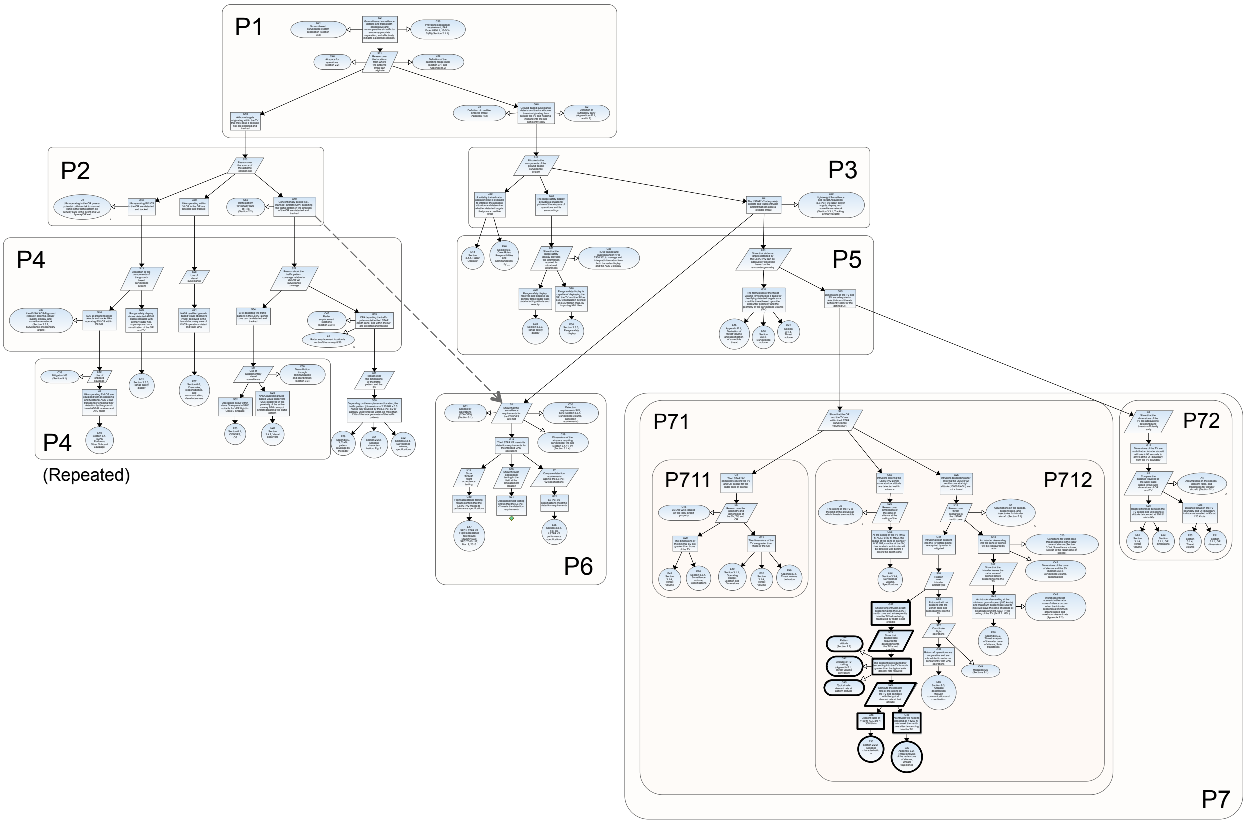


Figure 8 — Zoomed out view of the GSN argument structure for the surveillance function, conforming to its argument architecture. The identifiers of its architectural components (i.e., the argument patterns P1, P2, …) have been shown to illustrate the mapping.

The parent-child relation represents the order of argument fragments while the hierarchy relation is used to encapsulate successively larger components of the argument. Nodes contain the pattern name and are (optionally) annotated with the system components they apply to. For example, '*P2: Enumeration over type/origin of events*' applies to both UAs and general aviation (GA) aircraft. Thus, in the underlying safety argument, we will have two adjacent fragments, with similar reasoning, but applied to different artifacts. In comparison, '*P4: Allocation to surveillance system components*' is also applied twice, but at different levels. Thus, the argument fragment applied to the components *ADS-B Transceiver, Surveillance network* will be a child of the argument applied to *ADS-B, VO*, etc. The dashed arrow is used to represent additional dependencies. In this case, we observe that separate parts of the argument concern the same system component, suggesting a potential refactoring.

The underlying argument (shown in Figure 8, zoomed out and mapped to the patterns comprising the argument architecture) asserts that the surveillance function used, meets the operational safety requirement, i.e., that both cooperative and non-cooperative air traffic can be detected and tracked to ensure appropriate separation and effectively mitigate a potential collision. It is worth noting that this argument (and its architecture) are themselves fragments of a much larger argument for the overall system (not shown). Additionally, recall that the nodes in Figure 8 shown with the thick dark borders correspond to the GSN argument fragment presented earlier (Figure 3) in this paper. In general, there is not a unique relationship between an argument and an argument architecture; that is, a given argument need not have a single, unique argument architecture. Different architectures could be constructed to represent different aspects and omit various details. Moreover, an argument might contain bespoke, "glue argumentation", that is not represented at an architectural level. An example of this, which can be seen in the left *leg* of the structure of Figure 8, is the fragment of the GSN argument structure *not mapped* to a pattern in its argument architecture.

Relating the Argument and the Safety Architecture: In general, an assurance argument can address safety concerns, but those that are organized according to safety barriers should provide a rationale for why each control is effective, how it meets its requirements, why it prevents threat events from progressing to the top event, and generally explaining the context in which the barrier operates. Thus, we would thus have a set of arguments for each control, along with an argument that explains how they contribute to safety when combined. However, here we have not considered this *top-level* argument, and defer that to future work. Also, the arguments we have created currently do not consider escalation factors, which are effectively subsumed by an appeal to the reliability of relevant components of the controls.

Relating the Safety Architecture and the Argument Architecture: At a high-level we might expect there to be a close correspondence between the safety architecture and the argument architecture, i.e., we could map each control into an argument fragment explaining why that control works. Intuitively, each system component of a hazard control (e.g., radar) should map to a fragment reasoning about the requirements on that component, i.e., the architectural entity parameterized over that concrete component, while threat events and consequence events in the BTDs should map to fragments that justify their mitigation. However, since the arguments we have presented here are low-level, being concerned primarily with the effectiveness of the detection and surveillance controls, the correspondence is more distributed.

Conclusion

Previously (ref. 16), we proposed an integrated systems and safety engineering methodology for sUAS. There, the focus was primarily supporting the airworthiness assessment of the vehicle platform, i.e., the UAS itself. We have, since, extended that approach towards the wider safety case, developing a methodology for capturing the associated safety rationale in the form of structured arguments, showing an application to both airworthiness and operational safety (ref. 8). In this paper, we have further extended our prior work to provide two notions of architecture for presenting the key content of UAS safety cases: while the abstract safety architecture represents the safety controls and their relationships, the argument architecture provide a high-level justification for why those controls meet their safety objectives. Although, our notion of a collection of BTDs is *one* possible representation of the abstract safety architecture, and there may be other possibilities. These architectures can play distinct roles, looking at the safety problem from different perspectives, and we have found that each is useful in identifying omissions/inconsistencies in the other. We believe that this combination of structured argumentation and bow-tie modeling thus provides a novel framework for architecting UAS safety cases.

We have used our tool AdvoCATE (ref. 17) to construct the safety arguments presented here, which have formed part of safety cases prepared at NASA and submitted to the FAA. AdvoCATE has other features, not described here, including a query/view mechanism—useful for understanding safety arguments—and safety argument patterns, which provide a basis for argument architectures (ref. 18, 19). We are currently working to implement the techniques described here, in particular by extending the view capabilities to encompass support for BTDs, safety architectures, their various views, as well as architecture analysis, composition, and generation. Regarding the relation between argument, and safety architecture, thus far we have only developed the lower-levels of the safety argument (and its architecture), corresponding (in a sense) to specific controls of the safety architecture. As future work, we will develop the upper-levels of such arguments, justifying how the overall collection of safety measures work to provide an acceptable level of safety. This work grew out of our ongoing efforts in developing a series of safety cases (in part for NASA's UTM effort), where we have sought to find ways to facilitate the reuse of safety artifacts, and to improve comprehension of the safety case by diverse stakeholders. As these missions proceed, we will also continue to evaluate how these techniques help with both reuse and comprehension.

Acknowledgement

This work has been supported by the Safe Autonomous Systems Operations (SASO) Project, under the Airspace Operations and Safety Program of the NASA Aeronautics Research Mission Directorate.

References

1. Prevot, T., Rios, J., Kopardekar, P., Robinson III, J., Johnson, M., and Jung, J., "UAS Traffic Management (UTM) Concept of Operations to Safely Enable Low Altitude Flight Operations," *16th AIAA Aviation Technology, Integration, and Operations Conference*, AIAA-2016-3292, Jun. 2016.

2. US Department of Transportation, FAA, “Unmanned Aircraft Operations in the National Airspace System (NAS),” Air Traffic Organization Policy Notice JO 7210.891, Nov. 2015.

3. US Department of Transportation, FAA, “Flight Standards Information Management System, Volume 16, Unmanned Aircraft Systems,” Order 8900.1, Jun. 2014.

4. Berthold, R., Denney, E., Fladeland, M., Pai, G., Storms, B., and Sumich, M., “Assuring Ground-based Detect and Avoid for UAS Operations,” *33rd IEEE/AIAA Digital Avionics Systems Conference (DASC 2014)*, Oct. 2014, 6A1–1–6A1–16.

5. International Civil Aviation Organization (ICAO), *Convention on International Civil Aviation*, 9th ed., 2006.

6. FAA Air Traffic Organization, *Safety Management System Manual version 4.0*, Federal Aviation Administration, May 2014.

7. Denney, E. and Pai, G., “Argument-based Airworthiness Assurance of Small UAS,” *34th IEEE/AIAA Digital Avionics Systems Conference (DASC 2015)*, Sep. 2015, 5E4–1–5E4–17.

8. Denney, E. and Pai, G., “A Methodology for the Development of Assurance Arguments for Unmanned Aircraft Systems,” *33rd International System Safety Conference (ISSC 2015)*, Aug. 2015.

9. Airline Risk Management Solutions (ARMS) Working Group, *Methodology for Operational Risk Assessment for Aviation Organizations v. 4.1*, European Strategic Safety Initiative (ESSI), Mar. 2010.

10.FAA Air Traffic Organization, “Air Traffic Organization 2014 Safety Report,” 2014.

11. Williams, B., Clothier, R., Fulton, N., Lin, X., Johnson, S., and Cox, K., “Building the Safety Case for UAS Operations in Support of Natural Disaster Response,” *14th AIAA Aviation Technology, Integration, and Operations Conference*, Jun. 2014.

12. Clothier, R. A., Williams, B. P., and Fulton, N. L., “Structuring the safety case for unmanned aircraft system operations in non-segregated airspace,” *Safety Science*, Vol. 79, 2015, 213 – 228.

13. Goal Structuring Notation Working Group, “GSN Community Standard Version 1,” Nov. 2011.

14. Denney, E. and Pai, G., “A Formal Basis for Safety Case Patterns,” *32nd International Conference on Computer Safety, Reliability and Security (SAFECOMP 2013)*, LNCS Vol. 8153, Sep. 2013, 21–32.

15. Industrial Avionics Working Group, “Modular Software Safety Case Process GSN – MSSC 203 Issue 1,” Nov. 2012.

16. Denney, E., Ippolito, C., Lee, R., and Pai, G., “An Integrated Safety and Systems Engineering Methodology for Small Unmanned Aircraft Systems,” *AIAA Infotech@Aerospace Conferences*, AIAA 2012-2572, June 2012.

17. Denney, E., Pai, G., and Pohl, J., “AdvoCATE: An Assurance Case Automation Toolset,” *31st International Conference on Computer Safety, Reliability, and Security (SAFECOMP 2012),* LNCS Vol. 7613, Sep. 2012, 8–21.

18. Denney, E., Naylor, D., and Pai, G., “Querying Safety Cases,” *33rd International Conference on Computer Safety, Reliability and Security (SAFECOMP 2014)*, LNCS Vol. 8666, Sep. 2014, 294–309.

19. Denney, E. and Pai, G., “Composition of Safety Argument Patterns,” *35th International Conference on Computer Safety, Reliability and Security (SAFECOMP 2016)*, Sep. 2016 (to appear).

Biography

Ewen Denney, Ph.D., SGT, Inc., NASA Ames Research Center, Moffett Field, CA 94035, USA, telephone – (650) 604-2274, e-mail – ewen.denney@nasa.gov.

Ewen Denney is a Senior Computer Scientist in the Robust Software Engineering (RSE) technical area of the Intelligent Systems Division (Code TI), at NASA Ames Research Center, where he has worked on automated code generation and safety certification in the aerospace domain. He is a co-recipient of a 2014 NASA Group Achievement Award, and he holds a Ph.D. in Computer Science from the University of Edinburgh.

Dr. Ganesh Pai, Ph.D., SGT, Inc., NASA Ames Research Center, Moffett Field, CA 94035, USA, telephone – (650) 604-0760, e-mail – ganesh.pai@nasa.gov.

Ganesh Pai is a Research Engineer in the Robust Software Engineering (RSE) technical area of the Intelligent Systems Division (Code TI), at NASA Ames Research Center, where he works in the area of safety assurance as applied to unmanned aircraft systems, and aviation systems/software. Dr. Pai is a co-recipient of a 2014 NASA Group Achievement Award, and he holds a Ph.D. in Computer Engineering from the University of Virginia.