The Role of Safety Architectures in Aviation Safety Cases[☆]

Ewen Denney, Ganesh Pai[*], Iain Whiteside

SGT / NASA Ames Research Center
Moffett Field, CA 94035, USA

Abstract

We develop a notion of safety architecture (SA), based on an extension to Bow Tie Diagrams (BTDs), to characterize the overall scope of the mitigation measures undertaken to provide safety assurance at both design time and during operations. We motivate the need for SAs, whilst also illustrating their application and utility in the context of aviation systems, through an example based upon a safety case for an unmanned aircraft system mission that successfully underwent regulatory scrutiny. We elaborate how SAs fit into our overall safety assurance methodology, also discussing the key role they play in conjunction with structured assurance arguments to provide a more comprehensive basis for the associated safety case. We give a formal semantics as a basis for implementing both BTDs and SAs in our assurance case tool, AdvoCATE, describing the functionality afforded to support both the related safety analysis and subsequent development activities, e.g., enforcement of well-formedness properties, computation of residual risk, and model-based views and transformations.

Keywords: Argument structures, Assurance, Barrier models, Bow tie diagrams, Safety architecture, Safety case, Safety system, Unmanned aircraft systems

1. Introduction

Layered or barrier models of safety measures, as embodied by *Bow Tie Diagrams* (BTDs), have been used in civil aviation for operational safety risk management [2, 3]. In this approach, the safety measures being deployed are assumed to have already undergone some form of certification or acceptance so that there is a baseline for risk management. Consequently, the emphasis is largely on maintaining this baseline during operations through safety performance measurement, data-driven (operational) risk assessment, and by integrating the safety management system (SMS) [4]. BTDs are now also being adopted in the context of regulatory acceptance and operational approval of unmanned aircraft systems (UAS)—our main application domain for this paper—being recommended as the basis for the associated safety case, although they continue to retain their operational focus [5, 6]. In our own work [7], we have used BTDs to create UAS safety cases to support flight operations conducted as part of the UAS traffic management (UTM) effort [8] at the National Aeronautics and Space Administration (NASA).

Based on that experience, our observation is that an operational safety focus, although useful and essential, by itself only partially addresses the different facets of the safety case that must be provided to the regulator—for example, the assurance concerns that arise when introducing a new technical implementation of an existing safety function, or an entirely new safety function within an existing system. To provide a more comprehensive basis for an aviation safety case and, thereby, an integrated approach to through-life safety, we extend BTDs to the notion of safety architecture (SA). Moreover, we also use this extension of BTDs for design-time safety analysis, in addition to operational safety. The goal is to present the comprehensive collection of measures taken to eliminate, reduce, or control safety risk both

[☆]This article is an expanded version of an earlier paper [1] by the authors.

[*]Corresponding author.
Email addresses: ewen.denney@nasa.gov (Ewen Denney), ganesh.pai@nasa.gov (Ganesh Pai), iain.whiteside@nasa.gov (Iain Whiteside)

prior to, and during, operations, whilst substantiating safety claims with evidence, so that there is assurance that the system as designed and operated is adequate, appropriate, and effective.

Towards this goal, we have recently given a formalization of BTDs that generalizes to a notion of *safety architecture* (SA), using a model-driven approach [1, 9]. In this paper, we build upon and substantially extend this prior work by *i*) elaborating the essential role that an SA plays in an aviation safety case, also clarifying its relation to assurance rationale captured using structured argumentation; *ii*) further developing the underlying theoretical framework including significantly expanding the formal basis for risk quantification; *iii*) giving new details on tool support; whilst *iv*) using a detailed worked example to highlight the usefulness and application of the formal framework and the tool. Together with our work in [1], we make the following contributions:

1) We describe how BTDs fit into traditional hazard analysis activities recommended by the prevailing safety risk management (SRM) processes, e.g., as in [10, 11], with a view to facilitating their use to support design-time safety analysis and pre-operational safety assurance.

2) We formalize SAs as a structure corresponding to the composition of various BTDs of a system, to capture the *big picture* with respect to system safety. To our knowledge, bow tie modeling does not traditionally provide a representation or means for viewing the full scope of safety concerns. Thus, we reconcile those situations where hazards identified through traditional hazard analysis can be associated with one or more BTDs, each of which can themselves share different admissible BTD elements. Moreover, we clarify the relation between structural elements of SAs and safety risk analysis.

3) We define structural properties to maintain internal consistency across the whole assemblage of BTDs, when combining them into an SA. As we will see later (Section 4.5), arbitrarily combining certain legitimate BTDs can produce some structures that we may want to rule out. Although notions of *chaining* BTDs[1] (loosely related to the concept of SA) have been considered, so far as we are aware there is a lack of compatibility rules.

4) We give a notion of *views* to support specific activities in both the assurance and development processes[2], e.g., risk assessment, and specification of barrier functionality. The idea is, again, to enable the use of BTDs during design for pre-operational safety assurance, as well as to facilitate reuse.

5) We develop a simple approach for high-level risk quantification, which we illustrate with a worked example. The approach is based on views of a given SA, and it makes simplifying assumptions so that order-of-magnitude estimates can be given for the probability component of safety risk.

6) In general, there is a lack of support for integrating BTDs and *assurance arguments* within a common safety case. Based on the formalization of BTDs and SAs as a first-class notion within our toolset, AdvoCATE [9, 12], we discuss how assurance arguments can be associated with various elements of SAs.

7) We describe tool support in AdvoCATE for the above activities, showing how we construct an SA as a collection of BTDs, enforcing consistency between bow ties and structural well-formedness on individual BTDs. We describe how an SA is provided with an initial risk assignment, and how this is used to compute residual risk levels. Lastly, we briefly describe the model-based capabilities AdvoCATE provides for developing SAs.

Our paper is organized as follows: in Section 2 we give a background on BTDs and structured arguments, also highlighting related work. Thereafter, in Section 3 we describe our methodology for using BTDs and SAs, and how it fits within existing safety analysis processes. Then, to motivate and simultaneously exemplify the utility of the contributions of this paper, Section 4 gives a running example based on an actual UAS safety case that we authored. Section 5 gives a formalization for SAs along with the basis for risk assessment. Section 6 discusses how, by using specific views, we can undertake a high-level, semi-quantitative risk assessment to justify the achievement of safety targets. Here, we also elaborate how the integration of BTDs and assurance arguments underpin a UAS safety case. Section 7 describes the essential role of SAs in an aviation safety case, and concludes the paper identifying future research avenues.

2. Background

In this section, we first describe related work, followed by a brief overview of BTDs, a summary of structured arguments and their representation, and a description of our concept of (aviation) *safety case*.

[1] For example, see http://www.cgerisk.com/knowledge-base/risk-assessment/chaining-bowties

[2] That is, the development processes to engineer the mitigation mechanisms that comprise the safety system.

2.1. Related Work

As mentioned in Section 1, BTDs have been used in civil aviation and are also being adopted for safety assurance of UAS. As such, a number of commercial tools are available that offer bow tie modeling capabilities, such as BowTieXP[3], BowTie Pro[4], THESIS BowTie[5], RiskView[6], SafetyBarrierManager[7], etc. With the exception of one tool that does support argument development [13], to the best of our knowledge, no other tool besides ours [9, 12] provides a common framework to integrate BTDs with assurance arguments for UAS safety case development. Moreover, none of the above tools currently offer capabilities to create SAs, BTD views, or to enforce non-trivial well-formedness properties.

The implementation of SAs in our tool uses a model-driven approach [9], although those details are out of scope for this paper. Again, as indicated in Section 1, the current paper builds upon and extends our recent work on SAs [1]. The latter, itself, emerged from our more general approach of combining BTDs with structured assurance arguments [14]. The broad idea is that the former provides a foundation for risk modeling, analysis, and visualization, while the latter facilitates capturing safety rationale. Along these lines, we have advanced a notion of *Risk Informed Safety Case* (RISC) applicable to UAS [15] that addresses various safety-related assurance concerns using a tiered approach, leveraging structured arguments, and using BTDs for risk visualization. We have also previously explored combining assurance arguments with BTDs to support both pre-operational safety assurance—in particular, to address type design conformance and airworthiness [16]—and operational safety [17].

Our notion of SA is different from, but compatible with classical safety control architectures, e.g., single channel 1 out of 1 (1oo1), dual channel 1 out of 2 (1oo2), etc. These represent implementation-level organizations of safety instrumentation meant for achieving a specified *level of safety integrity* [18], and it applies at a lower-level of abstraction than our notion. In the literature, the work in [19] is most closely related to ours. That approach reconciles early architectural knowledge of a system—modeled using the Architecture Analysis and Design Language (AADL)—with traditional safety analysis. However, the focus is on (safety) system design and pre-operational assurance. Our approach is, again, compatible with this work although our notion of SA is more abstract. Thus, we can conceive of a framework where an SA, as we model it, can then be developed in greater detail using, say, the approach in [19], or other model-based systems engineering (MBSE) approaches, e.g., using systems modeling language (SysML). As we will see subsequently in this paper, our notion of SA retains its operational relevance and, thus, can be tied to the underlying safety management system (SMS); this is a key distinction between our work and that in [19].

2.2. Bow Tie Diagrams

Bow Tie Diagrams (BTDs), also known as *Bow Tie Models* (BTMs) represent a *barrier model* of safety, and provide a graphical approach to visualize and assess the risk scenarios associated with a given hazard. The main components of a BTD are:

- *Hazard*: A controlled activity, condition, or entity that reflects a normal or desirable aspect of the concept of operations (CONOPS). Note that this conception of hazard is compatible with, but subtly different from, the traditional notion of hazard [10, 11] in that it acknowledges that the normal operations of a system can be inherently hazardous, and therefore defines the context for SRM activities. Thus, it is when control over a hazard is lost that it can be a source of harm. As such, it can be seen as a description of a *hazardous activity*. For example, UAS operations near an aerodrome, or in terminal airspace, is a hazard.
- *Top Event*: An undesired system state, where there is a loss of control over the hazard of the CONOPS, or a hazard release. In fact, in general, a top event in a BTD corresponds to that which is traditionally designated as a hazard in a preliminary or functional hazard analysis. We develop BTDs around a single top event associated with an identified hazard. For the hazard example given above, a possible top event is a loss of separation from other aircraft.
- *Threat*: A possible direct cause/source of the top event. Threats can include possible failure modes. An example of a threat for the above example of a top event, is a malfunction in the navigation capabilities of the UAS.

[3] Developed by CGE Risk Management Solutions, URL: http://www.cgerisk.com/
[4] URL: http://www.bowtiepro.com/
[5] Developed by ABS Consulting, URL: http://www.abs-group.com/
[6] Developed by Meercat Pty. Ltd., URL: http://www.meercat.com.au/
[7] URL: http://safetybarriermanager.com/

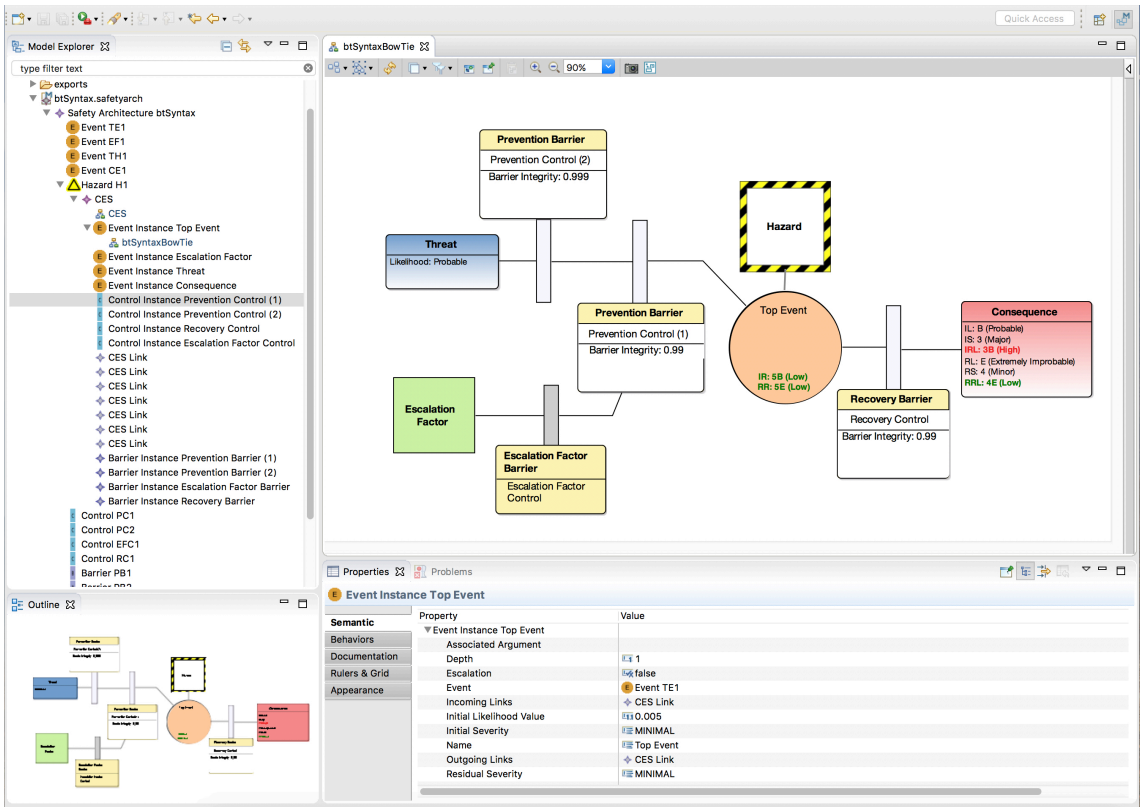


Fig. 1. Screenshot of AdvoCATE, showing the structure of an example BTD with its main elements: hazard, top event, threats, consequences, prevention and recovery controls, their Escalation Factors (EFs) and the corresponding Escalation Factor Barriers (EFBs).

- *Consequence*: The potential dangerous outcome or loss state resulting from the top event that must be avoided, e.g., serious or fatal injury to a third party. For the hazard and top event mentioned above, a key consequence to be avoided is a midair collision (MAC) with a conventionally piloted aircraft (CPA).
- *Control*: Any process, device, practice, or other action that modifies safety risk. An example of a control used when UAS operations are being conducted in civil airspace is a Notice to Airmen (NOTAM)—a notification filed with an aviation authority to alert aircraft pilots of potential danger along a flight route or at a location.
- *Barrier*: A collection or *system* of controls that contributes to reducing the probability of occurrence and/or magnitude of severity of the consequence(s) associated with a particular event within a chain of events describing a risk scenario. A surveillance system represents a barrier that is used to provide airspace users with awareness of the air traffic in a given airspace, thereby working to reduce the probability that aircraft will collide.
- *Escalation Factor* (EF): A weakness/vulnerability, threat, or operational condition that can compromise, defeat, or otherwise degrade control effectiveness. These can include environmental conditions, e.g., adverse weather, electromagnetic interference, etc.
- *Escalation Factor Barrier* (EFB): A *second tier* or secondary system of controls used to manage, reduce or modify the impact an escalation factor has on controls. In principle, an EFB is no different from a barrier, though it is placed at a lower-level in a BTD (See Fig. 1).

Fig. 1 shows a screenshot of AdvoCATE, illustrating a simple example BTD and its components. In general, a top event can have a plurality of threats and consequences, although Fig. 1 only shows a single threat and consequence for the top event. Intuitively, multiple threats and consequences connected to a single central top event can be seen to

resemble a bow tie, giving the structure its name.

In general, the BTDs can be interpreted as follows: barriers and controls to the left of the top event represent *preventative* mitigation measures, while those to the right of the top event are *recovery* measures to prevent the consequence from occurring. Preventative controls work to reduce the probability of the top event, whereas recovery controls contribute to reducing the probability of the consequence and/or the magnitude of the severity of the consequence, given that a top event has occurred. The visual ordering of barriers and controls corresponds loosely to the temporal order in which they may be invoked, in that they prevent the events preceding them, whereas events given after a barrier are interpreted to mean that the event occurs after the barrier has been breached. However, the diagrams (intentionally) abstract from the exact ordering and organization of barriers between successive events. Similarly, the ordering of controls within a barrier is not specified. Indeed, such groups of barriers and their constituent controls may operate sequentially, in parallel, in a continuous, or a demand mode, etc. These are design decisions which will be made after determining, at this abstract level, that the barriers being deployed can provide sufficient safety risk reduction.

We assume that threats are independently occurring events, i.e., there is a probability that threats can occur simultaneously and, therefore, they are not disjoint. Consequences may or may not be disjoint events. With this interpretation, threats, top events and consequences can be ascribed an *initial* and a *residual risk level*, computed as a combination of their (initial/residual) *likelihoods*[8] of occurrence and *severity*. Barriers and controls are each ascribed a measure of *integrity*, that relates to the probability that barriers are (not) breached in a dangerous manner. We adopt this term to distinguish it from the more familiar notion of *reliability*, which is concerned with *all* barrier failures or malfunctions. We also distinguish integrity and *integrity level*, where the latter represents a range of values for the former, corresponding to some factor or magnitude of risk reduction [18]. We will use these parameters in safety risk assessment.

Events, controls, and barriers can be used in multiple BTDs. Since distinct occurrences of events represent different contexts, we allow their likelihoods and severity to differ, referring to each occurrence as an *event instance*. Similarly, distinct *control instances* reflect the use of controls for potentially different threats, and can be assigned different integrities. Likewise, the properties of *barrier instances* depend on those of the constituent control instances; moreover barrier integrity also depends on the identified EFs and EFBs. However, when the distinction is not necessary we will generally refer to events, controls, and barriers, rather than their instances.

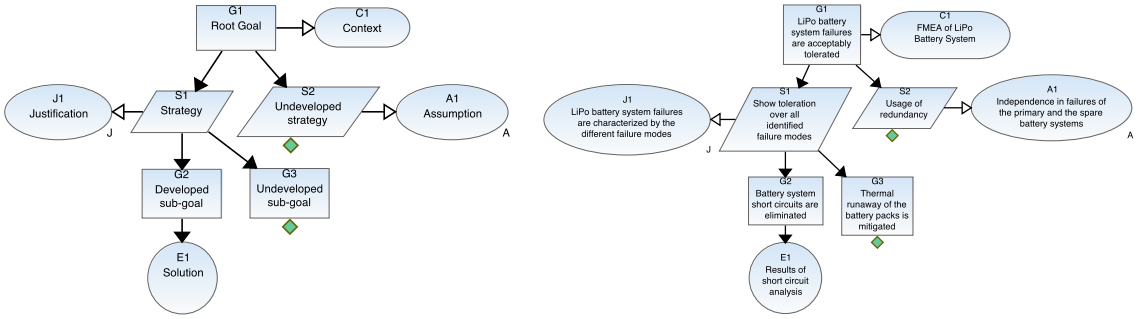
Note that a BTD can be viewed as a combination of a fault tree (FT) and an event tree (ET). For example, in one representation, the left half of a BTD is an FT such that the top event of the BTD is also the top event of the FT, and the right half of the BTD is the ET so that the top event of the BTD is the *initiating event* of the ET. Other mappings are also feasible [20]. BTDs can also be related to failure modes and effects analysis (FMEA) by viewing the results of the latter as providing insight into the various threats, EFs, and consequences that constitute a BTD.

2.3. 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*. Structured arguments can be graphically depicted as a directed acyclic graph of different nodes and links, e.g., using the Goal Structuring Notation (GSN) [21] as shown in Fig. 2. The *core* GSN (Fig. 2a) comprises six node types—i.e., *goals*, *strategies*, *contexts*, *assumptions*, *justifications*, and *solutions*—and two link types that specify, respectively, *support* (→) or *contextual* (⇾) types of relationships between the nodes. The GSN standard also includes notational extensions for modularity, though we will not cover those here. We also do not describe the steps or the methodology to create structured arguments in this paper, and refer interested readers to our previous work [14].

In general, nodes refer to external items including *a*) artifacts such as hazard logs, requirements documents, design documents, various relevant models of the system, etc., *b*) 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 *c*) records from ongoing operations, as well as prior operations, if applicable.

[8] Henceforth, unless otherwise stated, we mean *probability* when we refer to the term 'likelihood', and not the *likelihood function* in the Bayesian sense.



(a) Core GSN.

(b) Example GSN argument fragment.

Fig. 2. Graphical presentation of a structured argument using Goal Structuring Notation (GSN).

Fig. 2b is an illustrative example of a GSN argument fragment (retaining the layout of Fig. 2a to aid understanding) to substantiate the main claim (goal node G1) of acceptably tolerating failures for a Lithium-polymer (LiPo) battery system, in the context (node C1) of its Failure Modes and Effects Analysis (FMEA). The structure below node G1 elaborates the rationale for accepting the main claim. Specifically, the argument uses two complementary strategies, i.e., S1: showing that all identified failure modes are tolerated, and S2: using redundancy. The latter relies on an assumption (node A1) of independence in failures of the redundant systems, but has not been further developed, as indicated by the '◇' node decoration. The justification (node J1) for the former is based in the assertion that the different failure modes characterize the overall failure behavior. One of those failure modes concerns short circuits, whose elimination (node G2) is shown using the results of a short circuit analysis (node E1). Another failure mode pertains to thermal runaway, whose mitigation (node G3) is yet to be supported by evidence (again, indicated by the '◇' node decoration).

2.4. Safety Cases

A frequently referenced notion of safety case in the literature is that of "a structured argument, supported by a body of evidence that provides a compelling, comprehensible and valid case that a system is safe for a given application in a given environment" [22]. In this concept, structured arguments—which can often be of the kind described in the preceding section—play a core role for providing assurance, hence it is not uncommon to find "safety case" and "safety argument" being used interchangeably. However, informed by our practical experience in creating UAS safety cases, our observation is that often more information is required than that which can, or should, be captured by an argument. Moreover, so far as we are aware, there is limited guidance on when, or where a structured argument is appropriate.

Indeed, various aviation standards and guidance documents put forth other notions of safety case [23], [24], [25], [26], supplying their own context or application-specific interpretation for its exact purpose and nature, together with the required components, expected content, and presentation format. This may or may not include the requirement to present structured arguments. Although, these notions of safety case are largely similar to, and compatible with, those in which structured arguments play a central role. Nevertheless, we note that structured arguments are useful to *a*) explicitly trace safety considerations, from concept, to requirements, to evidence of risk mitigation and control, *b*) serve as a centralized organizing component of diverse assurance information.

For these reasons, we distinguish safety argument from safety case, and our concept of the latter considers the former as one among several core components—SAs, hazard logs, and an evidence repository being amongst the other core components—with the focus in this paper being on the SA component. Section 6 further develops how an SA contributes to the provision of assurance, while Section 7.1 elaborates the key role it plays in a safety case.

3. Methodology

Fig. 3 shows our high-level methodology for SRM to enable our goal of providing a more comprehensive basis for a UAS safety case. The rounded rectangular boxes give the activities performed, while the solid arrows indicate

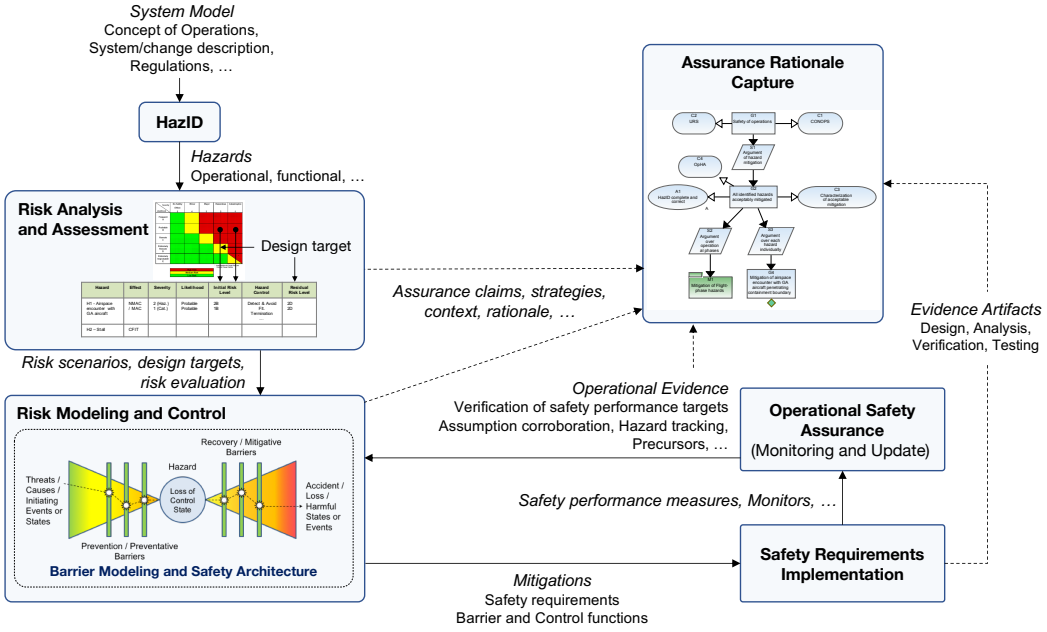


Fig. 3. Overview of our methodology for safety risk management in support of creating a UAS safety case.

the data produced and exchanged. The dashed arrows represent the information (and flow thereof) that pertains to assurance rationale, which we capture using GSN structured arguments.

In this paper, we are mainly concerned with the *Risk Modeling and Control* activity (see Fig. 3) where we create BTDs and the SA. We will also discuss how this relates to the activities of *Risk Analysis and Assessment* and *Assurance Rationale Capture*. The scope and details of the remaining activities, namely *Safety Requirements Implementation* and *Operational Safety Assurance*, are out of scope for this paper.

In practice, there is not a crisp separation between the three activities shown on the left in Fig. 3. In fact, the activities overlap and are iterative: identifying hazards often occurs in conjunction with safety risk analysis and assessment, as does safety risk modeling and the early identification of the risk control mechanisms. As such, there is a broad correspondence between our methodology and the generic SRM activities of hazard identification, safety risk modeling and analysis, and implementation and assurance (of the mitigation mechanisms), as shown in Fig. 4.

Next, we describe our methodology for developing a UAS safety case in the context of the generic SRM activities shown in Fig. 4, towards indicating how our approach is compatible with those recommended by the national aviation regulator [10, 11], as well as NASA [27].

Hazard Identification. Our methodology begins with an analysis of the UAS concept of operations (CONOPS), which describes the intended mission, the system usage, its boundaries, and its characteristics. The safety case and, in general, the supporting safety analysis must meet the relevant regulations and regulatory guidelines, such as those given in [23]. In our case, NASA requirements additionally apply, e.g., as specified in [27, 28]. Accordingly, we undertake a scenario-based hazard identification to create so-called *hazard risk statements* (HRS). As shown in Fig. 4, this corresponds to the activities labeled *HazID* and *Risk Analysis and Assessment*. That is, we elaborate the activities, conditions, or entities that pose a potential for harm, specifying the relevant operational context and system state, whilst also identifying various hazard consequences, with the focus on the worst-case outcomes. Traditionally, this has been documented in the form of hazard tables. Each such HRS can be mapped to the hazard, top event, and consequence elements of a BTD (Fig. 1). As mentioned earlier, from a bow tie perspective hazards capture operational contexts, whereas top events reflect loss of control system states that lead to harm if mismanaged or left unmitigated.

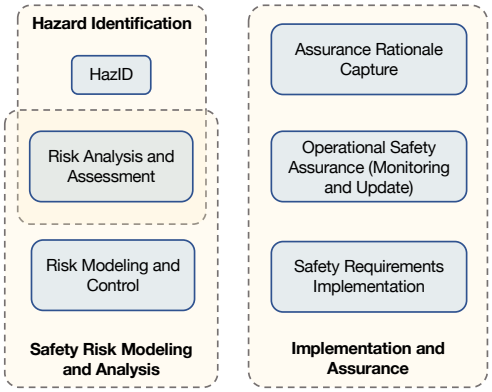


Fig. 4. Generic SRM activities—shown as the dashed boxes labeled *Hazard Identification*, *Safety Risk Modeling and Analysis*, and *Implementation and Assurance*—and their link to the activities of our methodology, showing how the two are compatible.

Safety Risk Modeling and Analysis. Next, we undertake a hazard analysis in parallel with incrementally developing BTDs, based upon which safety risk analysis and assessment is conducted. As indicated earlier, these steps are iterative and, as shown in Fig. 4 they broadly correspond to the activities labeled *Risk Analysis and Assessment* and *Risk Modeling and Control*. Here we give a high-level overview of these activities and their outcomes. Later (Section 4), we will give more details on the associated lower-level tasks, with the help of an illustrative example.

First we identify the events or situations that precede the scenario described by the HRS. Those are traditionally considered as *hazard causes* and in a BTD they correspond to the threats leading to a top event. Depending on the system hierarchy, some causes also will map to escalation factors (EFs). Then, we establish what pre-existing risk mitigation measures are available, and map those either to controls, barriers, or to escalation factor barriers (EFBs). The result is a preliminary collection of BTDs that provides the starting point for an initial risk analysis and assessment, helping to establish *initial risk levels* (IRLs) for the identified consequences. The IRL can be used to determine whether or not pre-existing safety mitigations are sufficient to provide an *acceptable level* of safety risk.

The basis for risk acceptance along with the corresponding *safety targets* are set in negotiation with the aviation regulator, and can be determined using a guideline such as [29]. In our case, NASA guidelines and requirements additionally apply. When the IRL is not sufficient, new mitigation mechanisms—i.e., controls and barriers—are required such that the resulting *residual risk levels* (RRLs) meet the required safety targets.

The assemblage of new and pre-existing mitigations can be for prevention and/or recovery depending on where they are situated in a given BTD. Depending on the level of detail to which mitigations measures are to be developed, we can refine the barriers on a specific path into their constituent controls, also including their EFs and EFBs. Intuitively—depending on system hierarchy, and how system boundaries are defined—we can map EFs and EFBs to the results of a preliminary or functional hazard analysis, as well as the results of lower-level safety activities such as preliminary system safety assessment (PSSA) [30], in much the same way as for top events, threats, consequences, and barriers. In other words, through a careful mapping of BTD elements to the results of existing safety analysis techniques, we can use BTDs at a pre-operational design stage when developing a system that requires (regulatory) approval, before it can be deployed into operation.

Thus, amongst the key outcomes from this process are various BTDs corresponding to the different HRS that, collectively, give the scope of UAS mission safety, and specify the measures to be undertaken for SRM. We can view this as a coherent and high-level picture of the overall *safety architecture* (SA), that describes how safety is designed into the system, and maintained during operations. In Section 6, we give more details on how we use the SA for risk analysis.

Implementation and Assurance. The next step is to develop safety requirements specifying the required, risk-reducing barrier/control functionality. As seen in Fig. 3, this is input for the *Safety Requirements Implementation* activity. In

fact, this activity is the point at which development processes for engineering barrier systems can take over, producing development artifacts including verification evidence that provides pre-operational assurance of barrier functionality.

Based on both the safety requirements, and their implementation, different safety performance measures can be developed together with monitoring mechanisms for the same. These facilitate *Operational Safety Assurance*, providing concrete operational evidence with which one can: *i*) verify that the barriers as specified meet their required safety performance targets, *ii*) corroborate and validate, or correct, any assumptions made, and *iii*) track the identified hazards (and top events), or detect new top events. This information serves as input to update the risk analysis and assessment. Fig. 3 also shows that operational evidence and development artifacts (produced from safety requirements implementation) collectively contribute to the evidence base used in *Assurance Rationale Capture*, the claims for which stem from the preceding hazard/risk modeling and analysis activities. In Section 6, we describe how SAs serve as an interface to address specific auxiliary assurance concerns. We will also discuss how and where structured argument are appropriate to capture lower-level assurance rationale—i.e., at the level of individual BTDs and specific bow tie elements (e.g., barriers)—in response to concerns such as barrier *fitness for purpose*.

Altogether, the SA can be seen to play a key role in SRM, and together with assurance arguments, it can be viewed as providing an underpinning for a (UAS) safety case. We refer the reader to our earlier work [14, 15] for details on the development of assurance arguments, and the role of rationale—captured in the form of structured arguments—in a safety case.

4. Illustrative Example

We now give an example—based on a safety case that we authored to obtain regulatory approval for conducting beyond visual line of sight (BVLOS) UAS operations as part of the NASA UTM effort [7, 8]—to illustrate the development and use of BTDs and SAs. From a methodological standpoint, the focus is mainly on the details of the *Risk Modeling and Control* activity (Fig. 3) as discussed in Section 3.

The CONOPS involves BVLOS operations with multiple small UAS, within a defined operating range (OR), a defined volume of airspace that encloses, for the most part, sparsely populated and minimally built-up areas on the surface. There are some urban pockets where the population density is large enough to be marked on an aeronautical chart. The air traffic within and outside the OR includes conventionally piloted aircraft (CPA), i.e., aircraft with onboard human pilots. The main safety concerns are to avoid harming non-participating (i.e., not involved in the CONOPS) third parties, and preventing property damage.

4.1. Initial Bow Tie Modeling

Following our methodology, we conduct a hazard identification based on the CONOPS, followed by a hazard and risk analysis, whilst creating BTDs in parallel. Fig. 5 (made using AdvoCATE) shows a fragment of one of the resulting BTDs, wherein one of the identified hazards, and its corresponding top event are:

- Hazard – *Airborne unmanned aircraft (UAs) operating BVLOS within the operating range (OR).*
- Top event – (E1) *Airborne conflict from a loss of separation*.

There are other top events associated with the hazard, such as a *deterioration of separation from terrain* (not shown here). A credible worst-case consequence for the identified hazard is (E2) *midair collision (MAC) between a UA and a non-cooperative*[9] *manned aircraft*. One of the main causes leading to the top event E1 is an airborne intrusion into the OR, which we have shown in Fig. 5 as the threat (E3) *non-cooperative aircraft intrudes into the OR when UAs are airborne*. There are also other threats (not shown here) that lead to the top event E1, such as *airborne excursion*, i.e., when UAS exit the OR.

Our next steps are to: *i*) identify pre-existing mitigations deployed in the wider air traffic management system that can be leveraged for risk reduction, *ii*) determine whether those mitigations are sufficient to establish that the risk posed is acceptable, *iii*) define new mitigation measures, if required, to further reduce risk to an acceptable level, and *iv*) if required and relevant, establish the EFs that can affect the barriers being used, along with the applicable EFBs. For the given CONOPS, pilot actions such as *see-and-avoid* (shown to the immediate right of the top event in Fig. 5) are an example of a particular pre-existing control prevalent in the current airspace system, that contributes to

[9] An aircraft not equipped with transponders, and not receiving air traffic services.

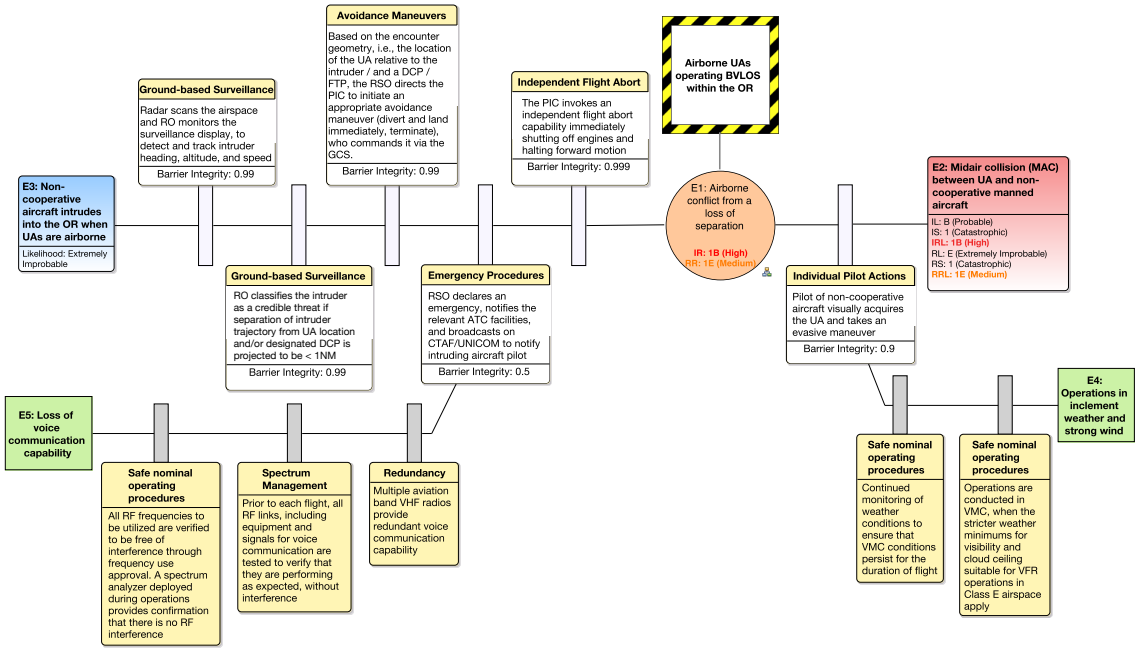


Fig. 5. Fragment of a BTD for our running example, showing the top event (E1: airborne conflict from a loss of separation), an identified threat (E3: intrusion into the operating range), and a worst-case consequence (E2: midair collision), together with specific risk mitigation controls/barriers, escalation factors (E4, E5), and their respective escalation factor barriers.

the safety function being provided by the containing barrier. That is, here the pre-existing barrier is '*individual pilot actions*' and the control required in this scenario is for the pilot to visually acquire any air traffic with which there is a conflict, and take an evasive action. Another example of a pre-existing barrier/control (not shown in Fig. 5) is the *communication and coordination* undertaken by air traffic control (ATC) to provide instructions to pilots when they have declared (and broadcast) an emergency. Yet some more pre-existing mitigations take the form of operational restrictions mandated by the regulator, e.g., limitations on the maximum operating altitude, the airspace class where flights are permitted, etc.

Fig. 5 also shows *barrier integrity*, as well as the IRLs and the RRLs for the top event (E1) and its consequence (E2). As shown, in our case, the IRL of the consequence is *High*. Thus, the prevailing mitigations are, by themselves, deemed insufficient to substantially reduce the risk of air (or ground) collision, thus warranting more safety analysis to establish additional barriers. A key new barrier that is required is an *independent flight abort* capability (shown to the immediate left of the top event E1 in Fig. 5) that acts as a last resort to prevent loss of separation, when all other barriers have been ineffective. Invoking a flight abort grounds a UA to reduce the chance of an airborne conflict and, potentially, a MAC. Other new barriers include *ground-based surveillance*, and *avoidance maneuvers* appropriate for the CONOPS and UA performance specifications.

Fig. 5 additionally gives some of the identified EFs (E4, E5) for the barriers and their corresponding EFBs. For instance, *loss of voice communication capability* (E5) is an EF to the *emergency procedures* barrier. If unchecked, E5 will preclude the ability of the range safety officer (RSO)—the crew member with the primary operational responsibility for safety of the UAS mission—to communicate during emergencies with either ATC or other pilots operating in the vicinity. EFBs that contribute to minimizing the associated risk include *redundancy* in the aviation radios used, and *spectrum management* to detect and minimize potential radio frequency (RF) interference.

4.2. Expanding the Scope of Analysis

We can now expand the scope of the BTD to discover opportunities to proactively manage hazards by considering precursors to the identified threats, or when the controls used do not suffice in terms of their effectiveness.

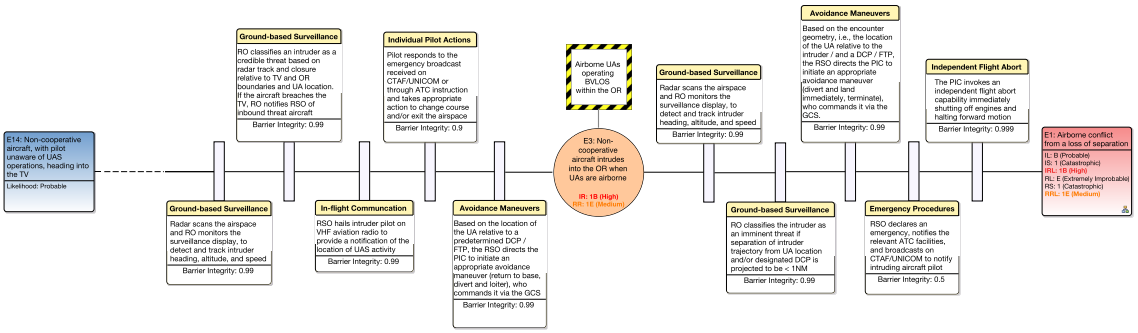


Fig. 6. Fragment of the new BTD created to analyze the threat E3 of the BTD of Fig. 5, due to which it is a top event here. The right side of this BTD fragment is identical to the left side of the BTD in Fig. 5. The analysis works to identify threats occurring earlier in the event chain and proceeds leftwards. The dotted lines indicate that there are other barriers besides those shown here to mitigate the new threat (E14).

Consider, for instance, the *ground-based surveillance* barrier in Fig. 5, shown to the immediate right of the threat of airborne intrusion (E3). Here, upon considering the characteristics of the airspace surrounding the OR, together with the worst-case assumptions on the behavior of the non-cooperative air traffic that may be encountered, a domain-specific analysis establishes that this control will not be sufficiently effective. In particular, we determine that classifying an intruder as a threat when the projected separation of its course from the UA position is less than 1 nautical mile (NM) affords an insufficient reaction time in the worst-case, to safely avoid an airborne conflict. To provide a sufficiently early warning, we derive new surveillance requirements, characterized in the form of a *threat volume* (TV) of airspace surrounding the OR [7]. That induces a change to the procedures for surveillance (i.e., scanning and tracking targets at a required minimum distance from the OR) and classification of the detected airborne targets, e.g., into *credible*, and *imminent* threats.

Accordingly, we further assess the identified threat (E3) of Fig. 5, by considering it as a top event in a different BTD. Within AdvoCATE, this creates a new, partially developed BTD (a fragment of which is shown in Fig. 6, such that the erstwhile threat (E3 in Fig. 5) is now designated as a top event, the previous top event (E1 in Fig. Fig. 5) is now a consequence (in Fig. 6), and all the barriers between the two (i.e., *ground-based surveillance*, *avoidance maneuvers*, *independent flight abort*, along with the applicable *emergency procedures*) are retained and re-used. Then, we repeat hazard analysis to identify additional threats, mitigating barriers, EFs, and EFBs as appropriate.

As shown in Fig. 6, a new threat—E14: *Non-cooperative aircraft, with pilot unaware of UAS operations, heading into the threat volume (TV)*—has been identified, capturing the contribution of a lack of awareness of other airspace users. There are other threats (not shown here), such as a lack of awareness of the UAS pilots themselves. Fig. 6 also reflects the modifications required to the controls of the surveillance barrier, which specifies how detection, tracking, and threat classification should (and will) occur. In particular, two additional barrier instances of *ground-based surveillance* that contain different controls are introduced between the threat (E14) and the top event (E3). Also, the control in the second (reused) instance of the surveillance barrier (appearing to the right of the top event E3) classifies airborne intruders differently than it did in the BTD fragment of Fig. 5.

These modifications and, in general, the expansion in scope of the analysis, results in the introduction of additional barriers—more precisely, specific additional controls of those barriers—i.e., *avoidance maneuvers*, *individual pilot actions*, and *in-flight communication* (all shown to the left of the top event E3, in Fig. 6), as well as *pre-mission communication and coordination*, *crew qualification*, etc. (not shown but suggested by the dotted line in Fig. 6).

Besides identifying and managing precursor events, as in the preceding discussion, we will often need to consider the same event in different contexts. For example, for the hazard considered in our running example, we can have different instances of a *ground collision* consequence event whose safety risk changes based upon where control is lost in the flight route, and where the collision occurs on the surface. We refer to these as distinct *event instances*, which share a description, but whose likelihood and severity values can vary.

At this point, we introduce a notion of *intermediate event* so as to allow us to show on a BTD path, an event that is neither a top event, threat, or consequence. Necessarily, an intermediate event lies between a threat and a top event,

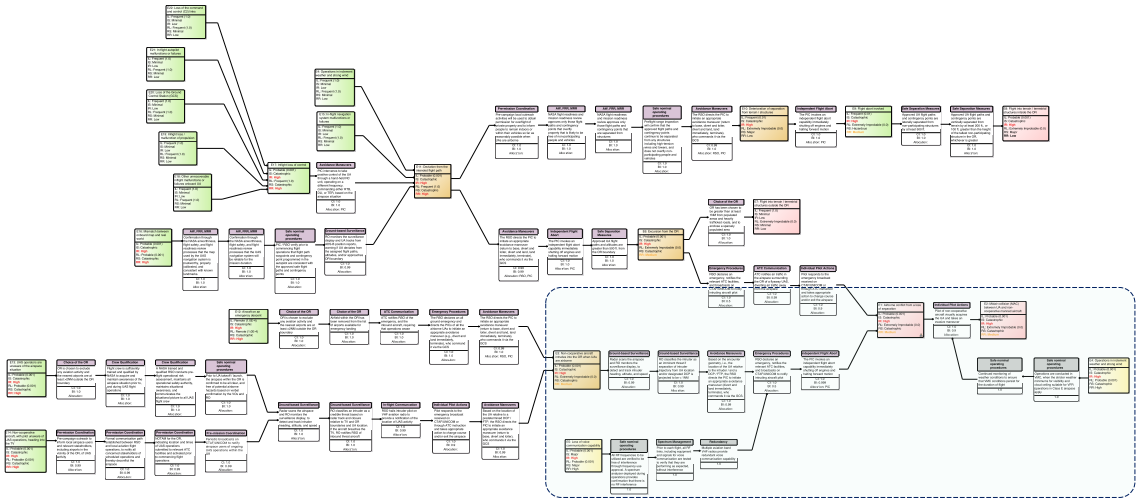


Fig. 7. Fragment of a partially developed safety architecture (SA) (shown zoomed out). The sub-fragment highlighted by the dotted lines corresponds to the BTD of Fig. 5.

or a top event and a consequence.

4.3. Towards a Safety Architecture

Generalizing this analysis and scope expansion, it is intuitive to see how we can incrementally develop different BTDs for each top event of each identified hazard. The resulting collection of BTDs comprises the SA for the system, and the overall structure can be seen to characterize the total scope of safety, describing various (operational) risk scenarios, and the applicable safety mitigations.

The need for an explicit SA concept and its visualization arises from the interconnections that emerge between different BTDs, which are not evident when considering them individually. For instance, for the CONOPS of our running example, there are BTDs for different top events of the same hazard, as well as BTDs where different hazards share the same top event (which will be distinct event instances). There is also flexibility in deploying barriers and controls, so that different BTDs can share event chains (an example of which are the BTDs of Fig. 5 and Fig. 6), and barriers can be reused for different threats not only in different BTDs, but also in the same BTD (on different paths). As mentioned earlier (Section 2.2), at this level of modeling abstraction, the exact order and organization of barriers between adjacent events is not a concern, although groups of barriers are ordered relative to the global temporal order of events in the SA.

Fig. 7 shows a fragment of a (partially developed) SA for our running example. Intuitively, this structure can be considered as the result of a *composition* of related BTDs [17]. However, the SA does not distinguish top events from threats or consequences, and we simply consider event chains along with the risk mitigation measures that work to stop the temporal progression of the events in a chain. The dotted box to the bottom right in Fig. 7 shows the part of the SA that corresponds to the BTD fragment of Fig. 5; the paths to its left are the events (threats) and controls resulting from expanding the scope of analysis as described earlier in Section 4.2. This dotted box can be seen as a moving window on the SA, that focuses the analysis in a BTD by considering a given event in the chain as the top event. In our model-driven implementation in AdvoCATE [9], the SA is automatically assembled in the background as the BTDs are being created. If required, however, the SA can be directly edited and the tool maintains consistency with the constituent BTDs. Our implementation relies on a metamodel (described in [9]) of SA and associated concepts, that closely follows the formalization that we will describe subsequently (in Section 5).

4.4. BTD Properties

As we develop an SA, it can be useful to highlight potential inconsistencies, and to check that certain well-formedness properties hold in the BTDs, i.e., those properties whose violations could translate into weaknesses in the

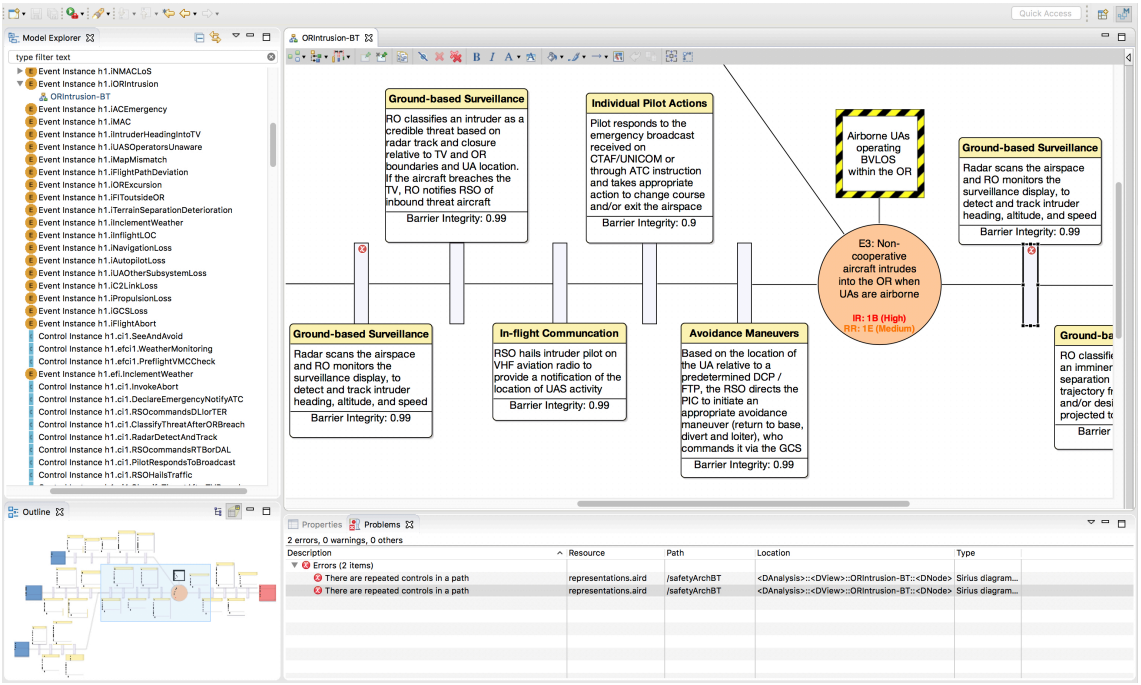


Fig. 8. Screenshot of AdvoCATE, capturing a violation of the property that controls should not be repeated on a path, as applied to the BTD of Fig. 6. The tool highlights the violation using error annotations '✘' on the applicable controls—in this case, of the *ground-based surveillance* barrier instances on either side of the top event (E3).

risk analysis and, as a consequence, in the implemented safety system.

For instance, if there are two paths t—**b**—c_1 and t—**b**—c_2, in different BTDs, where t is a threat (or top event), **b** represents a collection of barriers (or controls) and c_1, c_2 are from amongst different intermediate events, top events, or consequences, then there is a potential inconsistency in the overall SA if there are different outcomes for a common threat t and identical breaches of **b**. Such a situation may arise if there are missing barriers (or events) on the two paths. In some circumstances, though, we may want to allow both structures: for example, when c_1 and c_2 are, in fact, on the same causal chain but the respective BTDs are being applied at different levels of abstraction.

Using our running example, we now give two additional properties that AdvoCATE can check: the first applies to a single BTD, and the second arises due to the incremental treatment of the various hazards and the associated events.

4.4.1. Repeating Controls

Amongst the main principles governing barrier or control usage is *loose coupling* to reduce interdependence and reinforce defense in depth, so that unacceptable deviations in the safety performance of one or more barriers do not destabilize the rest of the safety system. This is violated when we repeat controls on a path, e.g., using the same control for prevention of a specific top event, as well as for recovery after that top event occurs. Thus, in the BTD of Fig. 6, repeating the same control of the *ground-based surveillance* barrier (concerning the actions of radar surveillance of the airspace and monitoring of the surveillance display) to both mitigate the threat E14 and to recover from the top event E3, violates the property of not using the same control in both a prevention and a recovery role on the same path.

AdvoCATE users are warned against this violation by highlighting the offending controls on the diagram, as shown in Fig. 8. A possible correction (not shown) could be to change one of the controls—e.g., the recovery control to be used after the top event E3—to indicate the exact surveillance action that will be undertaken as distinct from the control action taken before the top event E3. Thus, the former would involve surveillance of only those intruders

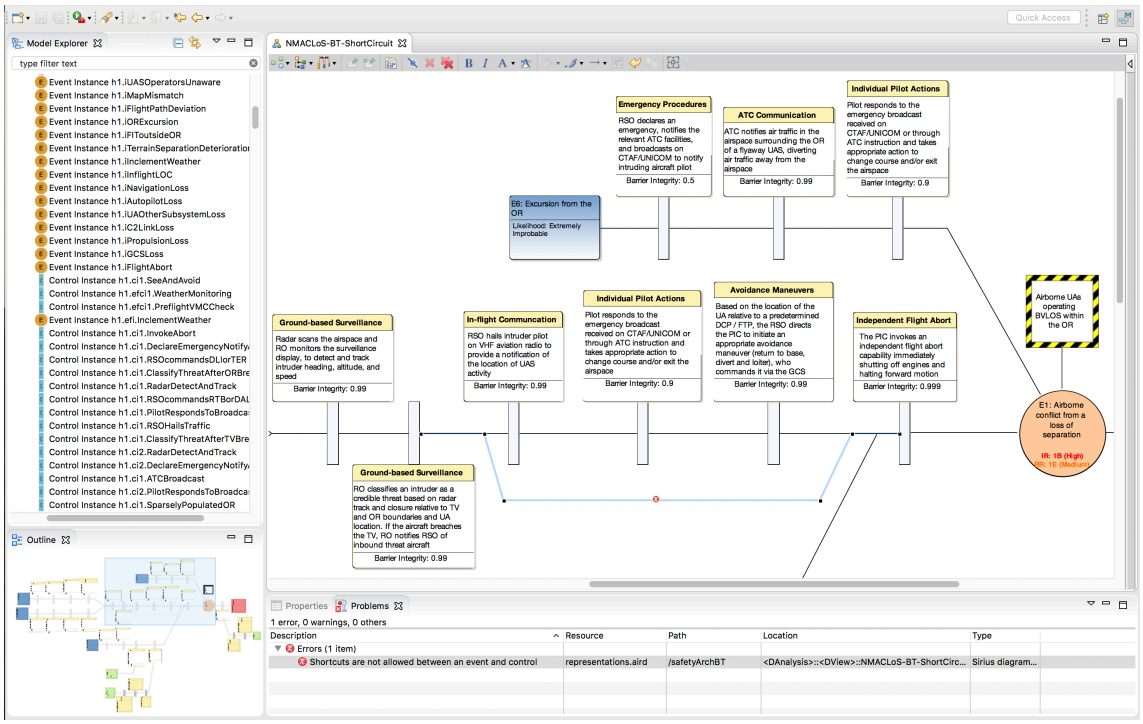


Fig. 9. Screenshot of AdvoCATE capturing a well-formedness violation of a *short-circuit* path between two controls (shown as the highlighted link with the error annotation '✘') resulting from the composition of otherwise well-formed BTDs with legitimate paths.

classified as *credible threats* and that are within the OR, whereas the latter would track *all* non-cooperative, non-participating air traffic outside the TV. Note that in some circumstances, and for specific event sequences where threats/top events may recur (e.g., in different mission phases), repeating the control may be warranted and the tool preferences can be changed to ignore the property and not raise a warning.

In our implementation in AdvoCATE, we enforce some properties by construction, e.g., consistency of event ordering, others raise *errors*, e.g., the presence of loops, whereas others are permitted with tool-supplied *warnings*. AdvoCATE makes a distinction between warnings, which are issues that the user ought to eventually address, but which should not impede progress, and errors, which will prevent some action (such as computing a residual risk level).

4.4.2. Bypassing Controls

In incrementally building up the SA—essentially, by composing well-formed BTDs [17]—there can be legitimate paths that bypass or *short circuit* controls and barriers [31]. That is, when some controls or barriers on a path are breached, (different) controls (of either the same or different barriers) subsequently used on that path are ineffective. Such structures can result when the composition attempts to reconcile different paths between the same pair of bow tie elements, such that there is at least one path with no other elements between the pair.

To illustrate, consider the BTD of Fig. 5 whose top event E1 is caused due to threat E3 that is, itself, a consequence in the BTD of Fig. 6. As mentioned in Section 4.2, and as indicated in Fig. 7, there are other threats earlier in this event chain, such as unaware UAS pilots, aircraft declaring emergencies, etc. Starting from these earlier threats, legitimate BTDs can be created with E1 as the top event where: *a*) in one BTD, the *independent flight abort* barrier is invoked well before the barriers between the events E14 and E3 as shown in Fig 5 and Fig. 6, or *b*) in another BTD, the *independent flight abort* barrier is not invoked, and neither are the barriers between the events E14 and E3. The rationale for creating such BTDs could be that the safety engineer is modeling the scenarios where the surveillance

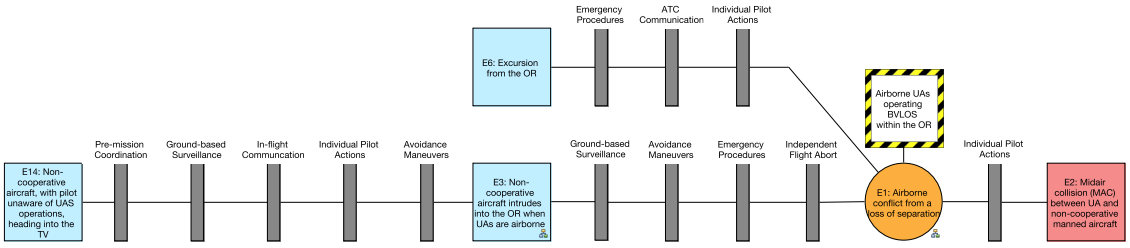


Fig. 10. *Barrier-centric view* for the running example, abstracting the BTDs of Fig. 5 and Fig. 6.

barrier is unavailable well before there is an airborne intrusion into the OR, or perhaps that there is a malfunction onboard the UA due to which commanded avoidance maneuvers are not executed.

If these BTDs (not shown) were then to be composed with the BTD fragments of Fig. 5 and Fig. 6, the result is an SA that contains short circuit paths. Although such paths should be prohibited in a well-formed BTD, AdvoCATE does create them also giving an error to indicate that there are, in fact, separate paths that may need to be reconciled—in this case between the relevant pair of barriers as shown by the highlighted link with an error annotation in Fig. 9.

To correct the violation, in the general case, either an intervening independent barrier is required on the (short circuiting as well as the short circuited) paths, or there are missing threats in at least one of the BTDs. In this specific case, we simply remove the short circuit path, since there already is an *independent flight abort* barrier that would be invoked either when surveillance were not to be available, or if avoidance maneuvers were to be unsuccessful. That is, in effect the possibility of those barrier breaches has already been *covered*.

4.5. Views

Even during the early stages of development, implicit or explicit choices may be required that affect the SA and, in turn, system safety. For example, choosing a specific type or number of surveillance sensors, using specific equipment onboard the airborne system, etc. Such choices are presented especially when deploying a system that is to be integrated—with or without changes—into an existing, wider system, as is the case for the CONOPS of our running example. The provision of different *views* of the overall SA can well support the associated trade-offs as well as other insights that can be used to further develop SA elements.

Moreover, as discussed in Section 3, risk analysis and assessment activities iterate with risk modeling and SA development. As we will see subsequently, views together with risk analysis can aid in providing early (semi-quantitative) assurance that the required safety targets can be met. In turn, that can be used to drive the subsequent development stages, e.g., by developing a high-level requirements specification for particular barrier functions. We now describe views that we have found to be useful in practice.

4.5.1. Barrier-centric View

One possible view of the SA shows a BTD with only the barriers shown, abstracting away the details of the specific constituent controls, and also consolidating repeated instances of the same barrier on the paths between two successive events. Fig. 10 is an example of such a *barrier-centric view* of the composition of the BTD of Fig. 5 (not considering the EFs or EFBs) and another BTD, a fragment of which is shown in Fig. 6. Besides the threats identified in those BTDs (i.e., E3, and E14), this view shows one additional threat (E6) labeled‘*Excursion from the OR*.’

The *simplified barrier-centric view* (not shown here)—which can be seen as the *classic* representation of a BTD [3], and which we create by a simplifying transformation not discussed here—is a variation on the barrier-centric view. It *i*) hides the intermediate events on a path, e.g., the threat E3 in Fig. 10; *ii*) splits converging paths, if any; *iii*) merges repeated barrier instances appearing on the same path, e.g., the instances of the ‘*ground-based surveillance*’ and ‘*avoidance maneuvers*’ barriers shown on the path between the threat E14 and the top event E1 in Fig. 10; and *iv*) does not merge repeated barrier instances that appear on different paths. In this transformation, we additionally require the barrier instances being combined to be of the same type, i.e., either prevention, or recovery, but not both. We combine separate barrier instances into a single barrier node in the view to indicate the overall contribution of that barrier to the global safety system.

Although these barrier-centric view variations abstract from the details of the controls being used, they are useful in a number of ways. Specifically, they give:

- A simple (semi-quantitative) way to assess the extent of risk reduction considering distinct barriers instead of their specific controls or barrier instances. As will be described in more detail in Section 6, going forwards from threats to top events to consequences, and assuming barrier independence, in this view we can determine whether or not the identified barriers collectively reduce risk to meet a safety target through a straightforward combination of threat probabilities and barrier integrity on the paths leading to top events or consequences.
- A higher-level of abstraction at which to apportion risk, given an *acceptable risk level*, or a safety target. Thus, going from consequences or top events to threats, the safety target can be apportioned across the various barriers. That, in turn, provides an integrity (reliability) requirement for barrier design. In this paper, we do not address the issue of risk apportionment.
- A graphical representation of the traceability from a specific hazard and top event to the barriers used to mitigate risk.

4.5.2. Slices

Another useful collection of views are *slices* relative to the different BTD elements. For example, a *barrier-centric slice* focusing on a specific prevention barrier gathers all the constituent controls, the threats being mitigated, and the top events that may result if the barrier is breached. Analogous to this is a barrier-centric slice where we present all the top events and consequence events related to a specific recovery barrier.

Fig. 11a shows an example of a barrier-centric slice across the SA of Fig. 7, focusing on the *ground-based surveillance* barrier. From the perspective of developing and implementing a barrier function, this view can be thought of as a specification of the required (prevention) functionality at a system level. Moreover, 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 component of the overall safety system—in this case, ground-based surveillance—is intending to address.

We can use this view to establish a simple metric of barrier or control *importance* based on the number of events associated with the slice (equivalently, controls within that barrier-centric slice). That can then be useful to determine the degree of assurance required. For example, the larger the number of events in the slice, the greater the importance, therefore potentially warranting a higher degree of assurance on barrier (and control) fitness for purpose, along with higher integrity. This view can also highlight potential issues with barrier use: for example, in Fig. 11a, the event E3 appears both as a threat being managed by, and the consequence of malfunction of, the same control. That suggests that the control needs to be refined to remove this circular path. In fact, this is exactly the control instance that violates the property of barrier or control repetition along a path (see Section 4.4.1).

Here, note that since the SA of Fig. 7 is partially developed, the barrier slice only shows those controls that have, thus far, been mapped to the barrier, along with the appropriate events being managed. The AdvoCATE implementation of views automatically updates this view (and others) as the SA is developed to completion.

Another type of a slice is an *event-slice view*, which presents all the hazards that are the context for that event, along with all its immediate precursor and successor events. In other words, this slice presents the scenarios and operating situations where the event in question occurs. Fig. 11b shows such an event-slice view for the event (E3) in the CONOPS, i.e., where a non-cooperative aircraft can intrude into the operating range. From this slice, we also see that E3 occurs in the context of another identified hazard—*UAs operating BVLOS in the OR along with participating CPA*—besides the hazard considered in the SA of Fig. 7.

Such a view could be useful to prioritize events from the standpoint of risk mitigation. Similar to the importance metric based on the barrier-centric slice view, we could establish an event *priority* metric, based on the number of hazards and the precursor/successor events that appear in an event-slice view. A variation (not shown here) to this view is a slice that focuses on a specific threat event, to present all the resulting successor (top) events and their related barriers. Effectively, that slice view shows the event chain across the entire system beginning from the threat under consideration. Such a view could 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 those threats. Analogously, a slice focusing on a consequence shows all the top events leading to a particular consequence/chain of consequences, the associated recovery barriers, and their organization.

In general, we can define other such slice views that focus on the different SA elements (or combination of elements), including EFs and EFBs.

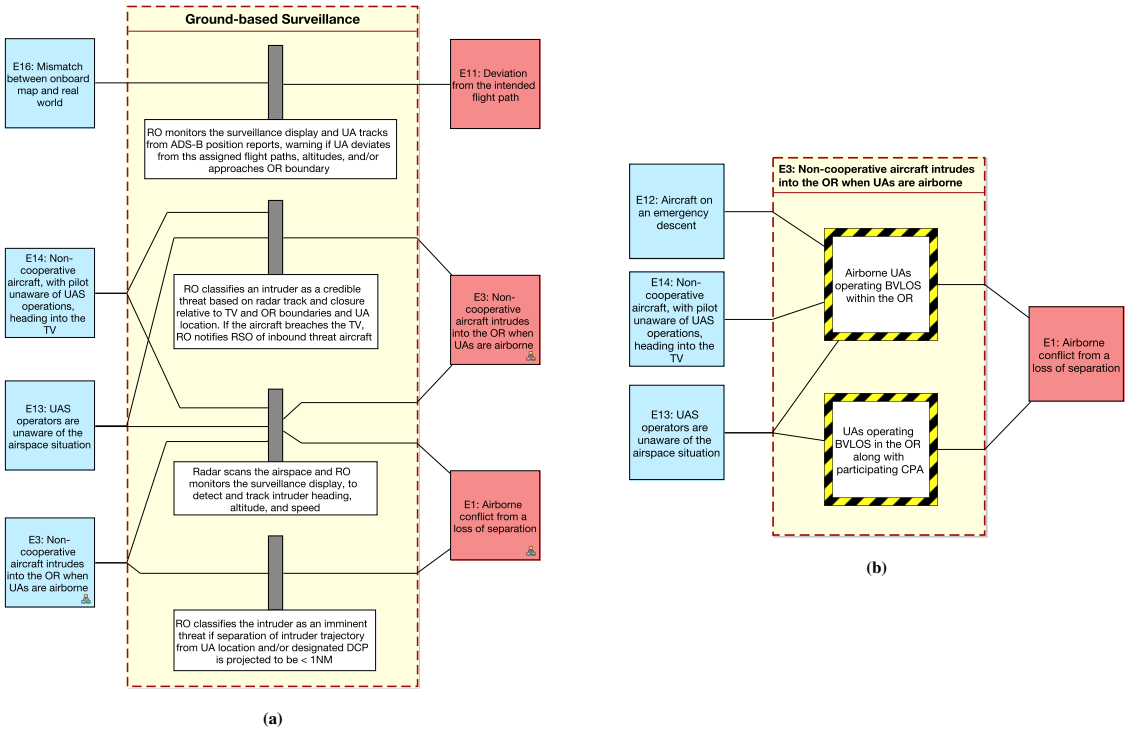


Fig. 11. *Slice* views: (a) *Barrier-centric slice view* of the *ground-based surveillance* barrier extracted from the SA fragment of Fig. 7, aggregating all the control instances and showing the events being mitigated, along with the events that result when the controls are breached. (b) *Event-slice view* for the event E3, extracted from the SA across all the identified hazards, along with the threats and consequences relative to that event.

4.5.3. Crew Allocation View

The *crew allocation view* (not shown here) is a specific type of non-graphical, operationally useful tabular presentation of the allocation from barriers (and the underlying controls) to specific crew members involved in the aviation operations. This view serves to provide a direct mapping from crew roles defined in the CONOPS, to specific responsibilities they will discharge in providing the safety function delivered by the associated (usually procedural) barriers. This view can help derive standard operating procedures, task checklists, etc.

5. Formalization

The worked example in the previous section illustrated some of the functionality that AdvoCATE, our assurance case tool, provides for creating and manipulating BTD and SAs. As can be seen, there are some subtleties about how distinct BTDs should relate to each other and what constitutes a legitimate SA. In order to provide a framework for clarifying these issues, we now develop a formal graph-theoretic semantics. This formalism serves as the basis for the implementation in AdvoCATE.

First, we note that the structures we want to define are parametrized over underlying sets of events, controls, and barriers.

Definition 1 (Safety Signature). *A* safety signature*, Σ, is a tuple $\langle E, C, B, bar \rangle$, where E, C, and B are disjoint sets of events, controls, and barriers, respectively, and bar : $C \to B$ associates each control with a unique barrier.*

We will henceforth assume the existence of a common safety signature for all definitions.

Before defining BTDs formally, we introduce the notion of *Controlled Event Structure* (CES), which represents the totality of all events associated with a hazard and their associated barriers and controls. In a sense to be made precise later, a collection of inter-related BTDs specifies this structure.

Definition 2 (Controlled Event Structure). *A CES is a tuple* $\langle N, \rightarrow, l, esc\rangle$ *where* N *is a finite set of nodes disjoint from the underlying safety signature,* $\langle N, \rightarrow\rangle$ *is a DAG representing temporal ordering of events,* l_x, $x \in \{t, d, c\}$ *is a family of labeling functions such that* $l_d : E \cup B \cup C \rightarrow$ *string gives descriptions and* $l_t : N \rightarrow E \cup B$ *gives node types. Writing* N_e *for* $\{n \in N \mid l_t(n) \in E\}$ *and similarly for* N_b*, we specify* $l_c : N_b \rightarrow \mathcal{P}(C)$ *such that if* $c \in l_c(n_b)$ *then* $bar(c) = l_t(n_b)$*, and* $esc : \{(b, c) \mid b \in N_b, c \in l_c(b)\} \rightarrow \mathcal{P}(N_e)$ *such that if* $c \in l_c(n_b)$ *and* $n_e \in esc(n_b, c)$ *then* $n_e \rightarrow^* n_b$*. Moreover, for all source and sink nodes,* x*, we require that* $l_t(x) \in E$*.*

We will use *CES* (Σ) to denote the collection of CESs over signature Σ. Definition 2 ensures that the links between controls and barriers respect the typing defined in the underlying signature.[10] Nodes of a CES represent specific events or barriers (and their associated controls) in the safety system. The same barriers can occur in multiple locations, though each may have different controls (and, as the next definition will make clear, can have different integrity). Each such location represents a distinct *barrier instance* and likewise for controls. Similarly, distinct nodes (in the same or different CESs) labeled with the same event represent *event instances*.

In contrast to BTDs, which are intended to represent (eventually) well-designed safety systems, a CES models a more general underlying set of events, with a partially developed safety system, possibly without controls yet in place. To allow more detailed modeling of a partially developed safety system, we also allow multiple intermediate events between controls, and allow events to have multiple successors (i.e., consequences) and precursors (i.e., causes), so that paths can split and rejoin.

We allow empty barriers since often we know we want a particular barrier before we have chosen or developed its constituent controls. Hence, the CES consists of a linked collection of events and barriers, rather than directly linking controls and events, with a map from a barrier instance to the (potentially empty) set of its constituent control instances (and not a sequence).

The interpretation of an escalation branch is that the barriers on the path between the escalation e and control c in barrier b represent escalation factor barriers. Note that the definition allows n-ary escalations, that is, escalations of escalation factor barriers, and so on, though in practice only a single level of escalation is typically modeled.

Definition 3 (Initial Risk Assignment). *An* initial risk assignment *for a CES consists of mappings* $int_b : N_b \rightarrow num$ *and* $int_c : N_b \times C \rightarrow num$ *(defined for all* (n_b, c) *where* $c \in l_c(n_b)$*), giving the integrity of barrier and control instances, respectively, and* $lik : N_e \rightarrow num$ *giving the initial likelihood of event instances.*

Thus we have a separate initial risk assignment for each hazard (i.e., the associated CES). We allow separate instances of controls and barriers to have distinct integrity values because they can have different effectiveness against different threats. Moreover, separate barrier instances can be implemented with different controls and are thus subject to different escalation factors. In practice, however, we currently assign integrity directly at the barrier (instance) rather than control level. Moreover, to simplify the calculation of residual risk levels, we amalgamate all barrier instances (see Section 4.5.1) so, in effect, use $int_b : B \rightarrow num$.

In general, we do not require any consistency between risk assignments for different hazards, since the context is different, though in practice there is likely to be overlap. We also assume the existence of a *risk classification* consisting of ordered sequences of *severity* and *likelihood* classes, along with a mapping $range : likelihood \rightarrow num \times num$. This mapping encodes an interpretation of a (discretized) risk matrix, giving the basis for computing IRLs and RRLs.

In fact, initial likelihood need only be specified for *global* threats, that is, those that do not have any preceding events. As we will see in the next section, residual likelihood is derived for all other events (working rightwards). We also assign severity to global consequences (i.e., those that are rightmost), and derive severity on all other events

[10] Alternatively, we could omit the signature and consider controls to be associated with barriers by virtue of their links, but the signature allows us to define controls in a barrier that are potentially but not currently used.

(working leftwards—the simplest approach is to set the severity of an event to be the maximum severity of its immediate successors). Given a *target level of safety*, we specify the *acceptable risk level* for individual consequences as a function from severity to likelihood, $l_{ar} : N_e \to (severity \to likelihood)$.

The integrity of a barrier is, in principle, derived from the integrities of its constituent controls. There are different ways of computing this and we will abstract away from the details here and assume the existence of an underlying *barrier risk model*, specified as follows:

Definition 4 (Barrier Risk Model). *A* barrier risk model*, comprises a set, R, and a semantic risk function such that for each control instance c, $[\![c]\!] \in R$, and functions $\mathbf{B} : R^n \to R$ (aggregating control risk semantics into a barrier), and $\mathbf{I} : R \to num$ (mapping the semantics of a barrier to an integrity).*

In this simple model barriers comprise no more than their constituent controls (abstracting, e.g., from their scheduling or pre-conditions). For a given model, additional constraints would relate barrier and control integrity (e.g., if a barrier has a single control it should have the same integrity value).

Example 1. *Assume a collection of Boolean random variables (r.v.) $R_b \cup R_c \cup R_e$, where $r_b \in R_b$ represents the breach (i.e., failure) of barrier b, $r_c \in R_c$ the failure of control c, and $r_e \in R_e$ the occurrence of escalation factor e. We define R to be the set of static (combinatorial) fault trees (FTs)*[11] *over RV. For those controls whose failure we can observe directly—i.e., controls without EFs—we simply assume the existence of FTs, $[\![c]\!] \in R$. This is the trivial FT with basic event r_c. Otherwise, for those controls with EFs, failure is defined using a FT whose elements, in turn, correspond to the EFs and EFBs. Thus, we can define $[\![c]\!]$ to be the disjunction of the FTs corresponding to the occurrence of each escalation factor of c, and the failure of all the barriers on that escalation branch. That is,*

$$[\![c]\!] = \mathrm{OR}_{i=1\ldots n}\left[\mathrm{AND}_{j=1\ldots m}\left(b_{ij}\right) \text{ AND } r_{e_i}\right]$$

where r_{e_i} is the Boolean r.v. for the i^{th} of n EFs, and b_{ij} is the Boolean r.v. for the j^{th} of m EFBs on the branch from the i^{th} EF. To aggregate the FTs corresponding to controls into those of barriers, we can define

$$\mathbf{B}\,(f_1, \ldots, f_n) = \mathrm{OR}_{i=1\ldots n}\,(f_i)$$

$\mathbf{I}(f)$ is then given by evaluating the FT f.

Here we assume that all controls and barriers are either directly assigned an FT or they are derived from those of simpler components. We also assume that controls within a barrier are independent in order to evaluate the FT in this simple way. More realistically, we should model the relations between the controls.

Currently, AdvoCATE users must specify the integrities of barriers directly and we will not consider this further here. In future we will support the use of external tools to implement barrier risk models. We can now define a safety architecture as a set of mutually consistent CES, over a common safety signature. Note that the factorization need not, in general, be unique.

Definition 5 (Safety Architecture). *Given safety signature, Σ, a* safety architecture*, $\langle H, l_h, ces\rangle$ consists of a set of hazards, H, hazard descriptions, $l_h : H \to string$, and a mapping $ces : H \to CES\,(\Sigma)$, which for each hazard, h, returns a controlled event structure such that the CESs are mutually temporally consistent, i.e., for hazards $h_1, h_2 \in H$, and events $e, e' \in E$, if $e \to_1^* e'$ then $e' \not\to_2^* e$ (where $\to_1$ and $\to_2$ are the temporal orders of the CESs of h_1 and h_2, respectively).*

Though we require mutual consistency for event ordering (in effect, that the combined relation on events is a directed acyclic graph (DAG)), we do not require similar consistency for controls between different CES since they represent different contexts in which controls can be used in different ways. This definition of safety architecture is slightly more permissive than the earlier one given in [17], but is more convenient for implementation. For example, one difference is that here we allow separate hazards to share top events. In a more detailed diagram, the events would be distinguished, but what really matters is that the combination of hazard and top event is unique. As we will show below, the definitions are essentially equivalent.

We will define a BTD as a specific kind of sub-structure of a CES.

[11] A static FT, a labeled tree comprising *events* and *gates*, models failure propagation paths as a Boolean combination of failure events [32].

Definition 6 (Bow Tie Diagram). *A* bow tie diagram*, B, is a CES such that: i) there is a designated event, e_{top}, called the* top event*; ii) there exists at least one* threat*, i.e., an event, e_t such that $e_t \rightarrow^* e_{top}$, and one* consequence*, $e_{top} \rightarrow^* e_c$; and iii) for all events e in B, one of the following holds: $e \rightarrow^* e_{top}$ or $e_{top} \rightarrow^* e$ or ($e \rightarrow^* c$ and $e_{top} \rightarrow^* c$) for some control c (i.e., an event is either a threat leading to the top event, a consequence following from the top event, or an EF of a control on a path to or from the top event.*

Note that EFs of prevention controls are subsumed by events on paths to the top event, whereas escalations of recovery controls must be explicitly accounted for in the definition.

In contrast to the classical notion of BTDs, here we permit arbitrary breadth BTDs with intermediate events and arbitrary depth escalations. Moreover, since we allow arbitrary event chains, paths can split and rejoin.

As mentioned in Section 4.4, a "good" BTD, however, will satisfy additional properties. For example, it is *free of short circuits* [31] if $e \rightarrow c \rightarrow e' \rightarrow c'$ implies $e \nrightarrow c'$. A stronger property that may be enforced is that no barrier can have multiple outputs. We say that a BTD is *maximal* relative to an SA if it includes all events before or after its top event, *well-controlled* if it has controls between all events (i.e., there is a barrier that contains at least one control) such that no control is repeated on the same path, and *well-escalated* if there is no control that appears both in some barrier, and in the escalation branch of another barrier on the same path. Other structural conditions that are indicative of poor design can also be checked.

Next, we show that a CES can be factorized into a set of mutually consistent BTDs which, in combination, give the original CES. To simplify matters, we only consider BTDs relative to a common parent structure, so they can be combined simply by merging (rather than by using a pushout as in [33], and introducing a notion of equivalence).

We also assume the existence of initial risk assignments. For shared threats, we must assume that initial likelihood is the same. Likewise, for severity levels of shared consequences. We need to take care with residual likelihoods since they can differ depending on whether they are computed over paths in a single BTD or the overall CES (see Section 6.1). However, since these values are derived, we can ignore them when considering the factorization and combination of BTDs.

Theorem 1 (Bow Tie Factorization). *A CES is equivalent to a set of mutually consistent BTDs for distinct top events.*

PROOF. Define an ordering on events: $e_1 \leq e_2 \iff \forall e \,.\, e_1 \sim e \Rightarrow e_2 \sim e$, where $\sim$ means that events are *comparable* (either $\rightarrow^*$ or $\leftarrow^*$). Intuitively, e_2 subsumes the set of potential threats and consequences of e_1. It can be seen that $\leq$ is a partial order, so since the set of nodes, N, is finite we can talk of maximal elements. Say that maximal events in $\leq$ are *central*.

Define a relation, R, on central events, by saying that t_1 and t_2 are *co-central* if they are $=$ relative to $\leq$. Equivalently, $t_1 \; R \; t_2 \iff t_1 \sim t_2$ and for all events, e, that are not between t_1 and t_2, if $e \rightarrow^* t_1$ then $e \rightarrow^* t_2$ and if $t_1 \rightarrow^* e$ then $t_2 \rightarrow^* e$. Since R is an equivalence (i.e., reflexive, symmetric, transitive) on central events (and a partial equivalence on all events), we can create the partition of central events in the CES by R.

Next, choose one member of each partition, and generate the maximal BTD (that is, the collection of all events comparable to that event, and the intermediate barriers, controls, and escalation branches) from it. This gives us a set of BTDs. They cover the CES, and can overlap, but are disjoint for central events (and top events, in particular). Since they are sub-dags of the CES they are mutually consistent. □

Thus, an SA can be thought of, equivalently, as a collection of (mutually consistent) BTDs for each hazard. The AdvoCATE implementation uses CESs, but users will typically view these as collections of BTDs.

6. Assurance Using Safety Architectures

Continuing with our running example (Section 4), we now describe the role of SAs (together with structured assurance arguments) in providing a more comprehensive assurance basis. Recalling the CONOPS, the safety case is concerned with showing that flight operations can be safely conducted, i.e., that *i*) a level of safety can be met that is equivalent to the required safety target (as negotiated with, or set by, the aviation regulator) for midair collisions (MACs), and *ii*) an acceptable level of safety risk is posed to the population on the ground.

The safety case is additionally required to show that a Ground-based Detect and Avoid (GBDAA) capability can be safely used in lieu of visual observers, the prevailing means of compliance with the federal aviation regulations (FARs)

deemed applicable to the CONOPS. The GBDAA system consists of a ground-based radar (with the provision for using multiple radars), transponders and supporting equipment, surveillance displays, along with a suite of avoidance maneuvers, and crew functions and procedures.

In the actual safety case that is the basis for the running example, we used the simplified barrier-centric view (see Section 4.5.1) of the SA created for the CONOPS to determine the level of risk reduction that would be achieved. That, in turn, was used to justify the claim that the safety system used would allow the operations to meet the target level of safety required by the regulator to grant operational approval. Claims about the fitness of purpose of the specific *new* barriers being deployed (i.e., *ground-based surveillance*, and *avoidance maneuvers*), were substantiated using GSN argument structures that marshaled evidence and reasoning from analytical models, operational and acceptance testing, and procedural details.

6.1. Quantifying Risk Reduction

To establish that the SA achieves an acceptable residual risk level (RRL) for the consequence(s), we use:

- Barrier integrity, given as nearest order of magnitude estimates based upon operational failure data, simulation data, and manufacturer specifications, where available, and/or conservative assumptions as appropriate.
- The (initial) probability of the threats;
- The (initial) severity of the worst-case consequence; and
- A *risk model*, based on the simplified barrier-centric view (see Section 4.5.1), that combines the above.

Unless there are barriers in the safety system whose specific function is to reduce the severity of a consequence (e.g., frangible airframes that do not lead to a hull loss upon impact), the worst-case consequence severity is the same as the initial severity when determining both the IRLs and RRLs. Thus, to assess the magnitude of risk reduction achieved, we mainly compute the reduction in the probability of the consequence(s) and the top events.

In our current implementation in AdvoCATE, the RRL, and in particular the residual probability of the consequence and top event, is computed, whereas the user specifies the initial probability (and thereby also the IRL). In principle, however, we can use the same approach for computing both the initial and the residual probability for the events under consideration, by selecting the appropriate barriers to include in the analysis. In other words, the IRL is dependent only on the pre-existing barriers, while the RRL includes the new barriers as well.

6.1.1. Formalization

Consider a BTD with a single threat T, top event $\mathbf{T}$, and single consequence C. The m prevention barriers[12] on the incoming path $\mathcal{I}$ from T to $\mathbf{T}$, are $\{P_1, P_2, \ldots, P_m\}$. Similarly, let the q recovery barriers on the outgoing path $\mathcal{O}$ from $\mathbf{T}$ to C be $\{R_1, R_2, \ldots, R_q\}$. We will abuse notation, letting T, $\mathbf{T}$, and C be synonymous, respectively, with the Boolean random variables (r.v.) modeling the threat, the top event, and the consequence respectively. Likewise, P_i (R_j) are Boolean r.v. modeling the prevention (recovery) barriers, while the Boolean r.v. $\mathcal{I}$ and $\mathcal{O}$ model the incoming and outgoing paths respectively, i.e., they model whether or not all the events on the path have occurred.

Except where otherwise stated, we use the following convention: $\Pr(X)$ is the probability distribution (mass) over all values (states) of the r.v. X, and $\Pr(X = x)$ is the probability that the discrete r.v. X is in the state x. When the r.v. is obvious from the context we will write $\Pr(X = x)$ as $\Pr(x)$. Also, $\Pr(x \mid y)$ is the conditional probability that X is in the state x given that Y is in the state y; and $\Pr(x, y)$ is the joint probability that X is in the state x and Y is in the state y. Since all the r.v. in our case are Boolean, they can model states such as $\{operational, failed\}$ (for barriers), and $\{occurs, does\ not\ occur\}$ (in the case of threats, top events, consequences, and escalation factors).

We now analytically derive the expressions to compute risk (probability) reduction. Let $\Pr(t)$ represent the initial probability that the threat T occurs, i.e., $\Pr(T = occurs) = \Pr(t)$. Similarly, $\Pr(\tau)$ is the marginal (i.e., unconditional) probability for the occurrence of the top event, and $\Pr(c)$ the marginal consequence occurrence probability. We let $\Pr(p_i)$, equivalently $\Pr(r_j)$, be the unconditional barrier *fragility*—denoting the opposite of barrier integrity—defined such that integrity of a (prevention) barrier P_i, is $1 - \Pr(p_i)$.

[12] Here we will consider barriers rather than their instances, so that there are only unique barriers on a path. The interpretation is that a barrier is compromised if any of its instances are compromised.

The probability of the incoming path $\Pr(\mathcal{I})$ is the joint probability $\Pr(T, P_1, \ldots, P_m, \mathbf{T})$ of all the events on that path, which we give using the chain rule of probability. Thus, we have

$$\begin{aligned} \Pr(\mathcal{I} = \mathit{true}) &= \Pr(t, p_1, \ldots, p_m, \tau) \\ &= \Pr(\tau \mid t, p_1, \ldots, p_m) \Pr(t \mid p_1, \ldots, p_m) \Pr(p_1 \mid p_2, \ \ldots, \ p_m) \ \ldots \ \Pr(p_m) \end{aligned} \tag{1}$$

To evaluate and simplify equation (1), we make two assumptions:

i) The barriers are—or should be designed, implemented and verified to be—mutually independent. Consequently,

$$\Pr(p_1 \mid p_2, \ \ldots, \ p_m) \ \ldots \ \Pr(p_m) = \Pr(p_1) \Pr(p_2) \ldots \Pr(p_m) = \prod_{i=1}^{m} \Pr(p_i) \tag{2}$$

This assumption may not always hold, in which case we must identify the dependent barriers (say, P_x and P_y) and consider the conditional probability that the respective barriers are breached (i.e., $\Pr(p_x \mid p_y) \Pr(p_y)$ or vice versa), determined through, for example, empirical means, simulation, etc.

ii) Barrier breaches are mutually independent of the threats (or top events). Recall that we determine barrier integrity, $1 - \Pr(p_i)$, using a barrier risk model (see Definition 4), which is based on the controls that constitute a barrier. Those, in turn, depend on the identified EFs and EFBs. Thus, this assumption amounts to assuming that when a barrier is breached, the threats (against which the barrier is being used) do not contribute to barrier inadequacy, and moreover that when a barrier is functional, it is perfectly effective against the threats for which it is used. Hence

$$\Pr(t \mid p_1, \ \ldots \ , p_m) = \Pr(t) \tag{3}$$

Again, in general this assumption may not always hold, since barrier effectiveness (and, thereby, its integrity) is situation dependent. In this case, we need to determine which barrier(s) are inadequate (say P_x) against the threat T and consider the Bayesian likelihood function $\Pr(p_x \mid t)$ that a barrier breach occurs due to its inadequacy against the threat. Then, using Bayes' theorem, we have

$$\Pr(t \mid p_x) = \frac{\Pr(p_x \mid t) \Pr(t)}{\Pr(p_x)} \tag{4}$$

which we use when reapplying the chain rule to evaluate the joint probability $p(t, p_1, \ldots, p_m, \tau)$.

From the semantics of BTDs, since the top event occurs when the threat occurs and all barriers are breached, $p(\tau \mid t, p_1, \ldots, p_m) = 1$. Applying this result, together with the assumptions in equations (2) and (3), to equation (1), we evaluate the path probability as the product of the initial threat probability and the fragility of all the barriers on the path. This is also the marginal probability that the top event $\Pr(\tau)$ occurs. Hence:

$$\Pr(\mathcal{I} = \mathit{true}) = \Pr(\tau) = \Pr(t) \prod_{i=1}^{m} \Pr(p_i) \tag{5}$$

Similarly, the probability of the outgoing path $\mathcal{O}$ from $\mathbf{T}$ to C, is the marginal (residual) probability of the consequence, $\Pr(c)$, which is given as:

$$\Pr(\mathcal{O} = \mathit{true}) = \Pr(\tau, r_1, \ldots, r_q, c) = \Pr(\tau) \prod_{j=1}^{q} \Pr(r_j) \tag{6}$$

In the general case, there can be w threats, $\{T_1, \ldots, T_w\}$, and n consequences, $\{C_1, \ldots, C_n\}$, for a top event $\mathbf{T}$, i.e., w distinct incoming paths $\mathcal{I}_i$ leading to $\mathbf{T}$, and n distinct outgoing paths $\mathcal{O}_j$ from $\mathbf{T}$. Then, the top event occurs if all the events occur on any given incoming path. That is, we evaluate $\Pr(\tau)$ from the probability of the disjunction of the w path r.v., so that

$$\Pr(\mathbf{T}) = \Pr\left(\bigcup_{i=1}^{w} \mathcal{I}_i \right) \tag{7}$$

We can compute equation (7) using the *inclusion-exclusion principle*. Since we require threats to be independent, the joint probability of a combination of threats occurring when evaluating this expression is simply the product of the individual threat probabilities.[13]

Since a BTD has exactly one path between a top event and any given consequence C_i, the residual probability of that consequence (in the given BTD) is the probability of that path, given by modifying equation (6) as

$$\Pr(c_i) = \Pr(\tau) \prod_{j=1}^{q} \Pr(r_{ij}) \tag{8}$$

where $\Pr(r_{ij})$ is the fragility of the j^{th} recovery barrier on the outgoing path from $\mathbf{T}$ to the i^{th} consequence. However, in an SA, there can be several distinct paths, each from a different top event to that consequence. If there are n such outgoing paths, O_j, then for a specific consequence C, we compute $\Pr(C)$ as the probability of the disjunction of the n path r.v. as,

$$\Pr(C) = \Pr\left(\bigcup_{j=1}^{n} O_j\right) \tag{9}$$

which we can compute using the inclusion-exclusion principle, as for the top event.

6.1.2. Application to the Running Example

We now illustrate an example risk assessment based on this formalization as applied to the running example. In particular, will use the *simplified barrier-centric view* (see Section 4.5.1), which applies a simplifying transformation to the barrier-centric view of Fig. 10 .

The generally accepted target level of safety for operations in civil airspace is 10^{-7} fatalities per flight hour. Using the qualitative risk acceptance matrix from [11] for risk classification, and the initial risk assignment function (see Definition 3), the safety target for the consequence (which has a *catastrophic* severity), is set as at least *extremely improbable*. That translates into an event frequency of the order of 10^{-7} MAC events per flight hour, with the assumption that a MAC results in a fatality. For other consequences that may have different severities, the acceptable risk level and the risk classification would determine the corresponding probability targets.

In the running example, given the nature of the CONOPS, a full fledged FMEA is deemed not feasible for the barriers being employed. Hence, a conservative assumption of the worst-case scenario is that for a given barrier any breach results in a safety related outcome, due to which barrier integrity is effectively the same as barrier reliability. Thus, for the rest of this section, when we refer to integrity, we are, in fact, using the value of (un)reliability.

Table 1 details the assumed integrity values, briefly detailing the rationale for the values chosen. Although, as we will see next (Section 6.2), we can use structured arguments to provide assurance of the integrity values, as well as to give a justification why barrier independence can be reasonably claimed. Due to a limited operational history of the barriers at the time of defining this SA, we are conservative in our assumptions for integrity, and assume that barriers are *frequently* compromised for the most part, i.e., qualitatively, ≥ 1 event per week $\approx 10^{-2}$ events per flight hour [11].

We refer to the specific paths in the barrier-centric view of Fig. 10 as follows:

- (NMAC | Pilot unaware) for the path from threat E14 to top event E1.
- (NMAC | OR exit) for the path from threat E6 to top event E1.
- (MAC | NMAC) for the path from top event E1 to consequence E2.

Then, based upon Section 6.1.1, and Table 1, we have:

$$\begin{aligned} \Pr(\text{NMAC} \mid \text{Pilot unaware}) &<< 10^{-7} \\ \Pr(\text{NMAC} \mid \text{OR exit}) &\approx \Pr(\text{E6}) \times 10^{-3} \\ \Pr(\text{MAC} \mid \text{NMAC}) &\approx \Pr(\text{E6}) \times 10^{-4} \end{aligned}$$

Here, we note that the eventual consequence probability depends on the initial probability of the threat E6 (*Excursion from the OR*), i.e., the UAS exiting the operating range. We are less concerned with the probability of a pilot not

[13] In general, however, threats may not be independent; e.g., when they relate to operational phases/modes. In this case, integration of phased mission analysis [34] and/or common mode analysis may be a plausible path forward, although we do not address it here.

Table 1. Assumptions and rationale for barrier integrity.

Barrier	Integrity	Rationale
Ground-based surveillance	$(1 - 10^{-2}) = 0.99$	Manufacturer specification of the expected time between failures for the selected radar system is ≈ 200 hours. Assuming an exponential failure distribution, the failure probability is of order of $\approx 10^{-3}$ for a 30 minute mission duration. We are further conservative and assume that overall unreliability is 10 times worse.
Avoidance maneuvers	$(1 - 10^{-2}) = 0.99$	COTS UAs do not undergo as rigorous an airworthiness determination and certification as aircraft operating under FAR parts 21, 23, and 25. UA systems including navigation, propulsion, autopilot, ground control station, and command and control (C2) links may fail/malfunction.
Independent flight abort	$(1 - 10^{-3}) = 0.999$	Nearest order of magnitude estimate, assuming a probability of failure on demand of 0.1 loss events per flight hour and a 30 minute mission duration.
Inflight and ATC communication	$(1 - 10^{-2}) = 0.99$	Redundancy is employed in voice communication radios for UAS pilots and ATC broadcast equipment is reliable. However, intruder aircraft may not be equipped with a radio, or pilots may not hear/respond to traffic advisory broadcasts.
Pre-mission coordination	$(1 - 10^{-2}) = 0.99$	Pilots may not consult NOTAMs or airspace users may willfully or unintentionally ignore UAS operations schedule.
See and avoid (Individual Pilot Actions)	$(1 - 10^{-1}) = 0.9$	Size, shape, and color of a given UA together with the encounter geometry may defeat the ability of a human pilot to visually acquire it early enough to avoid it more than once every 10 encounters.

being aware of UAS operations, i.e., threat E14, since it would be appreciably managed given the barriers in place. We observe that:

i) The threat E6 is dependent on the performance of the UAS, which undergoes an airworthiness determination prior to operations. Thus, the risk analysis suggests that an important aspect on which the SA (and airworthiness assessment) could potentially focus, is vehicle containment capabilities, i.e., *geofencing*. Other barriers that affect the probability of the threat E6 include flight route planning, direct UAS pilot intervention, etc.

ii) Since the analysis is based on the simplified barrier-centric view (see Section 4.5.1, and also Fig. 10), the view ought to be expanded further to include precursors to E6 and, thereby, include earlier mitigations. In fact, in the SA the *independent flight abort barrier* would be invoked prior to this event, but has not been considered in the specific view of Fig. 10). Since the integrity of that barrier is of the order of 10^{-3} (Table 1), the residual probability Pr(E6) is at least 10^{-3} for a view in which E6 is the consequence. Taking that value as the initial unconditional probability Pr(E6) in the present analysis, the path probability Pr(NMAC | OR exit) is of the order of 10^{-6} while Pr(MAC | NMAC) is of the order of 10^{-7}.

This reasoning can then be used to make the claim that safety architecture as modeled, in part, by the BTD of Fig. 10 is expected to meet the safety target, if the assumptions can be validated through operational evidence on barrier performance, e.g., from flight testing, simulation, etc.

6.1.3. Observations and Reflection

Although bow tie modeling traditionally de-emphasizes quantitative risk assessment (QRA)—focusing instead on the specific risk reduction measures to be used, and their links to the SMS—we have nevertheless used an underlying quantitative model. It is important to highlight (and we underscore the observation) that any QRA, especially in the context of safety, must be carefully, cautiously, and conscientiously interpreted [35].

Thus, we acknowledge that it is not practically feasible to verify the example risk analysis described earlier, or the achievement of the safety target. Moreover, the overall uncertainty in the analysis is too large for it to be the sole basis upon which to make a risk acceptance decision. However, by working through a quantitative model, nuances emerge that catalyze one to reflect upon the SA; for example, which barriers require more assurance, and ought to be the target of more rigorous development. In particular, it encourages clearly and explicitly stating the assumptions being made. It also focuses the safety engineering effort on specifying safety performance requirements for the barriers, such that there is a pathway to corroborate the collection of claims, assumptions, and development-stage evidence, using measurable operational evidence. In effect, it serves to reinforce the overall purpose of using the SA, which is to provide greater rigor in creating the safety case towards justifying the risk reduction claimed. As such, our approach to risk analysis is only the starting point for developing a mature QRA and, in its current form, can be characterized as a *qualitative risk analysis informed by quantitative thinking*.

We also note that it is indeed feasible and practicable to verify those barrier integrities that can be reasonably quantified. Moreover, since the SA is employing defense in depth with loosely coupled barriers, the individual integrities (as in Table 1) are of a magnitude such that measurement is feasible and can be established through statistical methods. For example, a hardware-only independent flight abort system could undergo accelerated life testing to verify that its integrity (in terms of unreliability) is at least of the order of 10^{-3}, together with a quantification of the uncertainty in the estimate. Moreover, by defining the appropriate safety performance measures [2], our approach can be made compatible with other data-driven aviation risk modeling approaches, for example, by following the modifications described in [36].

As mentioned in Section 4.5.1, the risk assessment is undertaken at the level of barriers rather than controls, since the abstraction provides a simpler way to compute the overall level of risk reduction. However, the trade-off is reduced accuracy. With the choice of an appropriate formalism, e.g., Bayesian networks and/or dynamic event/fault trees [20], the model could be made more accurate, accounting for dependencies (such as between controls within a barrier), and updating risk assessments using observed data. Nevertheless, this may not always be feasible in a number of situations: *a*) when crew operations are an intrinsic aspect of the overall safety system; *b*) providing conditional probabilities for certain event sequences, especially those associated with dependent control/barrier breaches; *c*) when controls or barriers are implemented largely, or entirely, in software; and *d*) when quantifying event probabilities that are inherently small, and with large variance in their estimates.

As a compromise that may be acceptable, we have explored combining BTDs and the SA with structured assurance arguments with some success in practice. The idea is that the (admittedly large) uncertainty in the risk estimates computed as described here, is (qualitatively) offset by stating and substantiating main and auxiliary safety-related claims through rationale. Indeed, such a framework can allow justifying certain conservative claims about barrier integrity, similar to the approach in [37].

We note that the safety case which has motivated our running example, was successfully evaluated and approved by the Federal Aviation Administration (FAA), the US national aviation regulator. That safety case marshaled not only the SA based risk analysis, but also other diverse evidence. Nevertheless, views of the SA and the risk analysis provided an intuitive way for the FAA—as well as the NASA Airworthiness and Flight Safety Review Board (AFSRB)—to better comprehend[14] how the CONOPS would deploy and manage defense in depth.

6.2. Relating BTDs to Assurance Arguments

BTDs and assurance arguments provide complementary assurance information. Broadly, the former can be viewed as providing the data required to instantiate patterns of the latter. For example, the core rationale why a BTD provides assurance of safety, is risk reduction through the application of diverse risk modification mechanisms. This can be

[14] Although our account is anecdotal, it is based upon actual feedback we received after detailed reviews by both the FAA and the AFSRB at NASA Ames Research Center.

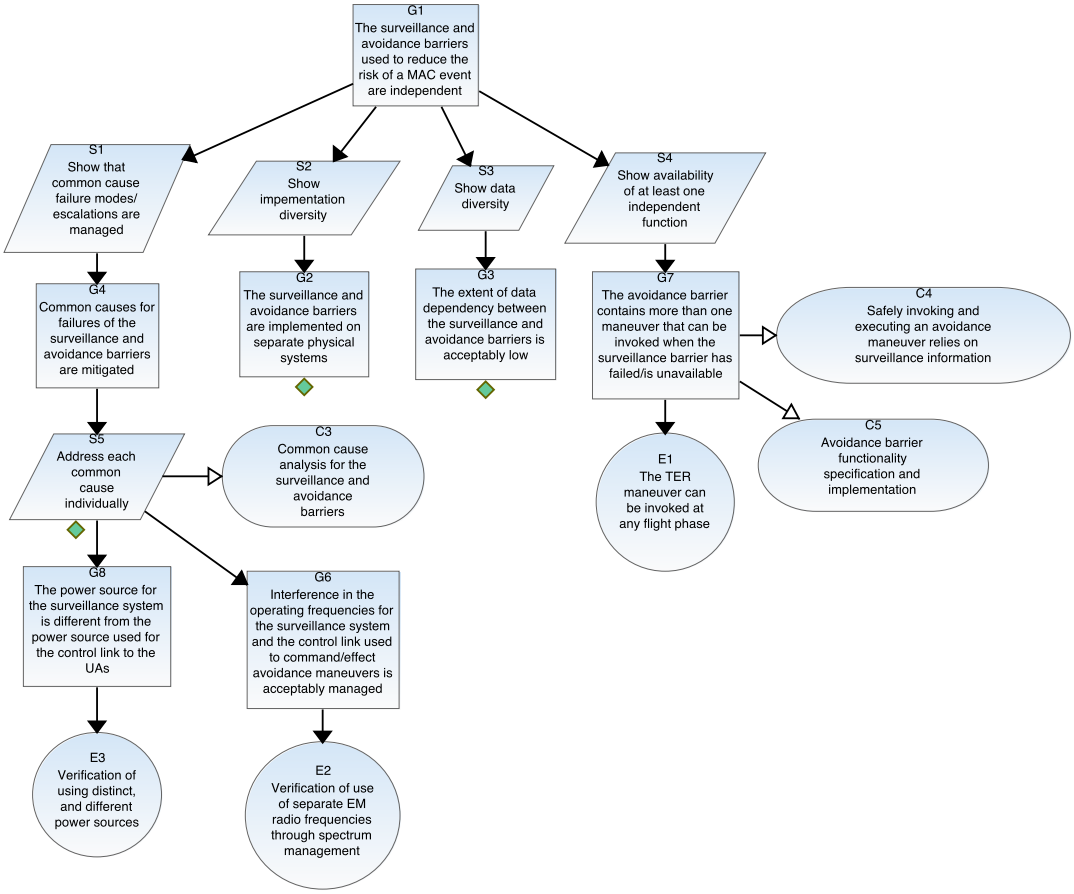


Fig. 12. Fragment of GSN argument for claiming independence between the *ground-based surveillance* and *avoidance maneuvers* barriers, used the running example.

captured as a structured argument, via argument patterns [33] that encode this reasoning, instantiated using the data from the corresponding BTDs. Conversely, we can associate a plurality of assurance arguments with a single BTD, multiple elements of a single BTD, or multiple BTDs, and more generally, the SA, where each argument addresses a specific assurance concern of the same.

For instance, an assurance argument corresponding to the BTD of Fig. 5 (or, equivalently, its barrier-centric view shown in Fig. 10) can be created, in which the main assertion is the reduction of the risk ascribed to the identified hazard (i.e., its top event) and its consequence. This argument would provide the rationale and evidence for how the barriers (especially those whose integrities have not been given in Table 1) contribute to reducing risk to an acceptable level. Auxiliary parts of this argument could involve, among other concerns, a detailed justification of barrier independence, sufficiency of the identified threats, etc.

Indeed, to provide assurance that the barriers in Table 1 are independent (with respect to their inability to deliver the required service), the argument could, for example, appeal to the usage of different, loosely coupled physical systems, their deployment and usage at different times, and evidence of little to no (known) safety-relevant data dependencies. Additional measures such as redundancy, and verification of non-interference of electromagnetic frequencies also would be used as evidence of having addressed certain common-cause failure modes. That, in turn, would provide added (qualitative) confidence in the overall claim of independence in barrier failures.

The rationale for claiming independence in the *ground-based surveillance* and *avoidance maneuvers* barriers is

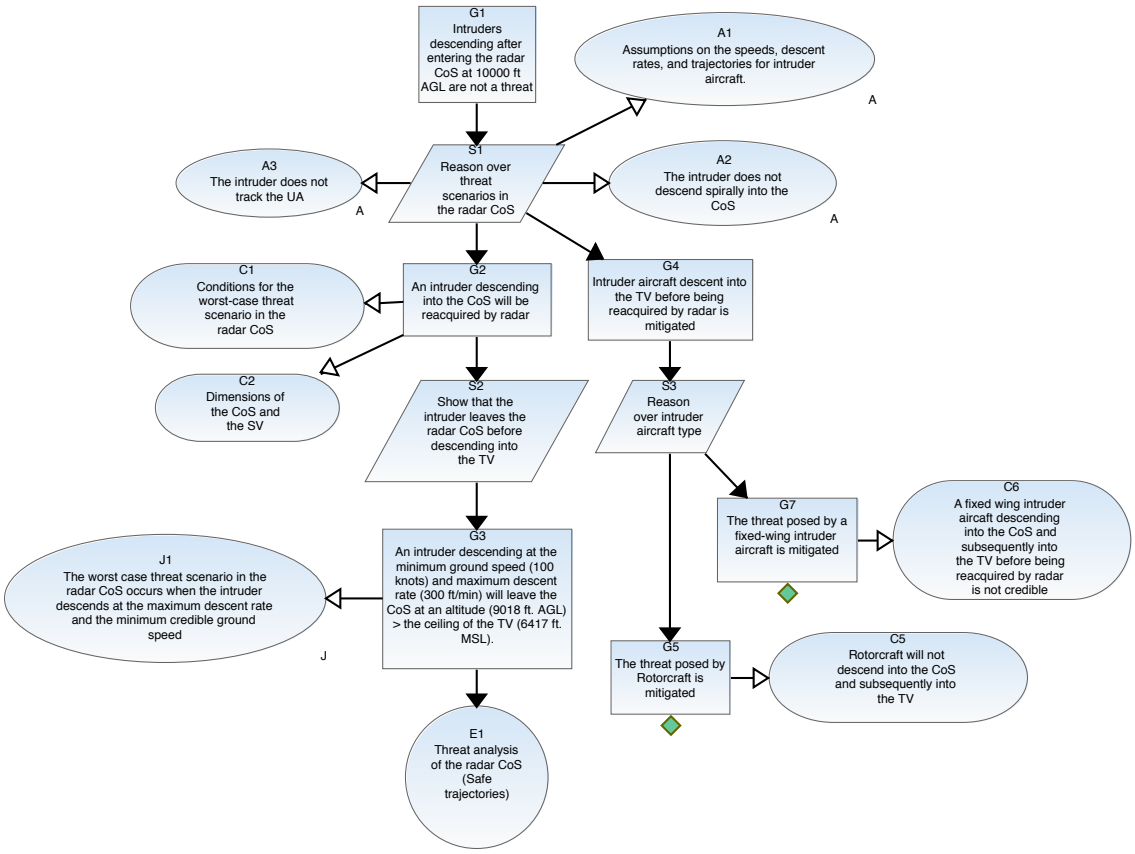


Fig. 13. Example GSN argument fragment providing rationale for why airborne threats in the radar cone of silence have been managed. This fragment is part of the broader argument that the ground-based surveillance system is fit for purpose, i.e., is capable of reliably detecting and tracking both cooperative and non-cooperative aircraft as required.

subtle, since the latter ought not to be realistically invoked unless the former is available and can be relied upon. Nevertheless, the suite of avoidance maneuvers is required to contain at least one maneuver, e.g., *land immediately* or *terminate*, which can be invoked irrespective of whether or not surveillance is available. The ability to rely on this maneuver when demanded, is primarily established through a vehicle-specific airworthiness determination which is, as a process, itself independent from the process used to establish that the surveillance system is fit for purpose. On the basis of this collective rationale, shown as the GSN argument fragment in Fig. 12, independence amongst these barriers also can be reasonably claimed for risk assessment.

Fig. 13 shows another GSN argument fragment of concerning a claim about intruder behavior in a region of the airspace where the surveillance system is *blind*[15]. That, in turn, is used in the assurance argument for the fitness of purpose of the surveillance barrier.

In this way, we can associate arguments with specific barriers and/or their constituent controls, wherein the top-level claims assert the provision of the required safety functions, a specific level of integrity, etc., and where the argument structure assembles detailed rationale and substantiating evidence. At a mission level, assurance arguments can also straddle a collection of BTDs, asserting how the overall SA enables safety in operation. For our running example, such an argument would relate to the SA of Fig. 7, addressing the overall scope of mission safety.

[15]The *radar cone of silence* is a conical region of airspace immediately above the radar that cannot be scanned due to the antenna elevation angle and mounting geometry.

From a methodological standpoint, to relate BTDs and assurance arguments we refer back to Fig. 3, in particular, the link between the activities of *risk modeling and control*, and *assurance rationale capture* (shown as the dashed diagonal arrow between the two activities). As shown, BTDs provide data that relate to the core elements that comprise assurance rationale. This includes *assurance claims*, e.g., pertaining to risk reduction, or hazard mitigation; *strategies*, e.g., appealing to the use of multiple layers of risk modification; *assumptions*, e.g., of independence between barriers, etc. In summary, structured arguments can be used to convey assurance *i*) for the wider system, by using assurance information embedded in the architectural concerns of the safety system as embodied by SA and its constituent BTDs. *ii*) at a lower level, for the properties of components of the safety system itself. This can be generalized to a notion of *tiered assurance* (not in scope for this paper, see [15] for additional details).

7. Concluding Remarks

7.1. The Role of Safety Architectures in Safety Cases

In this paper we have developed the notion of safety architecture (SA), based on Bow Tie Diagrams (BTDs), to provide a more comprehensive basis for assurance, especially when used in conjunction with structured assurance arguments. We now summarize its essential role in the provision of assurance in an aviation safety case.

Effectively, an SA describes various scenarios that lead to safety related consequences, together with the scenario-specific invocations of safety mitigation mechanisms. From the standpoint of design-time safety analysis, an SA on the whole provides a blueprint for the deployment of the overall *safety system*, while its various views (see Section 4.5) provide additional support during design. For instance, the barrier-centric view (Section 4.5.1) supplies the core rationale for how an SA—when properly implemented—will reduce risk, i.e., via defense-in-depth through independent, loosely-coupled, layers of protection (that is, barriers). This view also gives a basis for assessing the risk reduction achieved (Section 6.1), thereby supporting design choices and the associated trade-offs, e.g., by facilitating the evaluation of how risk is modified when there are changes to the safety system, such as the introduction of a new safety function, or the replacement of one barrier system by another. The risk assessment additionally serves as an interface for data driven quantitative risk assessment, and to *close the loop* on safety assurance. That is, by providing a model based upon which certain types of safety performance measures can be developed (e.g., those related to barrier integrity), to collect sound quantification data. Other types of views, such as a barrier-slice view (Section 4.5.2), are relevant for development of the barrier systems and their functionality, serving as a high-level functional safety specification. Since an SA is developed from a composition of BTDs, the latter can continue to be used in the traditional manner, i.e., during operational SRM.

As indicated earlier (Section 2.4), we distinguish the notion of safety case from that of safety argument, considering the latter as among the core components of the former, with SA being another of the core components. An SA provides assurance, through an integrated and consistent view on the full scope of the applicable safety concerns, while argumentation facilitates rationale capture to provide assurance. Together, they form two core components of a safety risk management methodology that supports *i*) pre-operational, development-time assurance required for regulatory acceptance of both potential changes to an existing safety system, and the introduction of new safety functions; as well as *ii*) operational safety assurance.

These facets constitute the core value addition provided by the extensions described in this paper, over other approaches.

7.2. Tool Support

We have applied this methodology in practice, in supporting safety analysis and safety case development in the context of unmanned aircraft systems (UAS), with its most recent use enabling BVLOS operations as part of the NASA UTM effort. Our work has leveraged AdvoCATE, our assurance case tool, using its functionality for creating and analyzing assurance artifacts—i.e., BTDs, SAs, and assurance arguments—as well as for property checking, creating views, for navigating between BTDs and arguments, etc. AdvoCATE also provides support for other steps of our SRM methodology, in particular the creation of hazard and requirements tables, and their traceability to the other components of assurance cases.

In practice, the tool support that is currently commercially available for creating barrier models (see Section 2.1 for more details) largely permits creating a disconnected collection of BTDs. To the best of our knowledge, they neither

support the creation of an SA as we have described it, nor do they provide view-based abstraction. In other words, none of the commercial tools provide the extensions we have developed in this paper, and our practical experience leads us to conclude that the functionality and methodology we have described requires tool support for them to be useful in practice.

The formalization we have described in Section 5 underpins the implementation of BTDs, SAs, views, and the support for risk assessment in AdvoCATE. We have employed a model-driven approach for implementation [9]; the corresponding models which have been implemented using the Eclipse Modeling Framework [38], closely follows the formalization we have described. Our overarching goal in formalizing SAs has been to provide a sound basis for implementing a model-based approach to BTDs, whereby multiple related BTDs can be represented consistently, and various formal views and properties computed. We have not chosen (at this point) to develop a fully formalized model. In particular, although we formally model the relations between controls and barriers we do not, currently, formally model the controls themselves though doing so would enable more precise risk quantification, at the expense of significantly increased modeling effort.

7.3. Future Work

We envision several lines of future work, each broadly contributing to improving the basis for assurance.

Tighter Integration of Safety Artifacts and Structured Argumentation. A broad vision is to integrate and provide support for the various safety analysis steps besides those related to the SA; maintaining the associated models and artifacts, including BTDs, safety requirements, the hazard log, associated implementation-related artifacts, assurance arguments and the related evidence; and ensuring consistency across all the linked safety artifacts.

Our ongoing work is investigating the relationship between assurance argument structures and BTDs both from *i*) the perspective of formal mappings that can be used to generate one from the other, and *ii*) how they best complement each other in a safety case. For example, one possibility is to associate argument patterns with generic controls, composing patterns to form an argument architecture, analogously to how controls are combined to form the safety architecture [33], and then instantiating the patterns based on the context in which the controls are used, to generate the associated assurance argument.

Through-life Safety Assurance. We have also developed a notion of *dynamic safety case* (DSC) [39] in our earlier work. An important avenue for future research is to explore how the SA can both inform and be the target of monitoring and update activities in DSCs. The rationale, as observed by [4], is that arguments are static and relatively inflexible especially when the safety focus shifts from acceptance for release into service, to operational risk management. Since the foundation of SAs are BTDs, which have traditionally largely supported operational safety, we submit that the SA would be well suited to link to safety performance measures, and the associated monitors.

Queries, Views and Transformation. We have currently implemented a fixed set of views, but plan to develop a more generic capability where views are generated from *queries* [40] and can be directly edited, maintaining consistency with the original diagram via bidirectional transformations. One such view, for example, could be a view that distinguishes new controls from the pre-existing ones, aggregating each into the appropriate barriers, to visualize how an existing safety system will be modified into a new one.

Moreover, we plan to investigate how views can be used to generate requirements on barriers. Along these lines, the barrier-slice view could be useful to synthesize checklists and standard operating procedures for both nominal operations and emergency situations, since this view aggregates all of the relevant procedural control actions.

Related to this, our model-based implementation [9] of the innovations in this paper also provides for preliminary model transformations that permit splitting/combining barriers and events in different ways, e.g., sequential versus parallel event split. From a modeling standpoint, these types of *refactoring transformations* are useful during incremental development, and we plan to investigate and develop more of them.

Hierarchy and Modularity. Similarly to how views abstract from concrete details, other abstraction mechanisms including modularity and hierarchy in BTDs are interesting future research avenues. Splitting an SA into separate BTDs can be seen as a natural form of modularity, as can grouping controls into barriers, and barriers into their categories. Individual diagrams can, nevertheless, grow large and in this case, hierarchy can be a useful tool for

size and complexity management. The AdvoCATE implementation already provides a form of showing and hiding different levels of BTDs and SAs.

Extending and Improving Risk Analysis. We have described a simple high-level risk analysis using a simplified barrier-centric view of the SA. One straightforward avenue of future work is linking to, being able to invoke, and integrating the results of more sophisticated models for quantitative analysis; for example, the combination of dynamic fault trees (FTs), ETs, and Bayesian networks [20].

Likewise, these models will also provide the mechanism to relate EFs, EFBs, and controls to barrier integrity. In turn, that modeling effort would benefit from extending bow tie elements with additional information such as an ordering on controls, and structural dependencies (e.g., sequential and parallel organization). We hypothesize that this would provide a common framework for modeling and analysis of SAs at multiple levels of abstraction, for instance, using our notion of SA at a high level, and at a lower-level—i.e., for the constituent barriers/controls—using the classical notion of safety instrumented system architecture (e.g., single channel 1oo1, dual channel 1oo2, etc.).

Presently the view-based simplification of the SA that we use for risk analysis cannot account for changes in the risk profile as a system moves from one operational phase to another. Thus another aspect of refining our risk analysis is to provide for *phased-mission analysis* [41]. This can be particularly useful in the context of UAS operations, since the risk profile changes not only during mission phases (e.g., during take-off, en-route, landing, etc.) but also within a phase itself, e.g., flight through different airspace classes, overflight of areas of different population density, etc.

Eventually, we aim to provide a basis for risk apportionment, and deriving the related safety performance requirements, together with tool support for the same. We additionally aim to support design choices, comparison and selection of safety mitigations, and the associated trade-offs, not only by extending the risk quantification model as indicated above, but also using techniques such as sensitivity and importance analysis [42].

Enrichment of the Model and Tool. As described earlier (Section 5), we have made some simplifying design decisions in the BTDs and the SA for implementation (e.g., providing barrier integrity directly), while other simplifications are inherent in the notion itself (e.g., abstracting the details of control ordering). We plan to close those gaps by extending the model to include additional information: for example, additional fields to capture additional barrier/control attributes such as effectiveness and adequacy, capturing a minimal form of control ordering and dependencies so as to better qualify the barrier risk model and the semantic risk function (see Definition 4), etc.

The latter is closely related to not only our plan to improve SA based risk analysis (by including formalisms such as FTs and ETs), but also improving the capability of AdvoCATE to integrate external analyses, and to export the analyses it creates to other external tools.

Supporting Practical Application and Reuse. Ultimately, we aim for our methodology and tool to be practically useful to a variety of end-users including safety engineers, development engineers, independent safety assessors, safety managers, regulators, etc. Towards this, there is a need to enable reuse, for example, through support for access to, and storage, in a *library* structure. Our intent is to be able to pre-populate a hazard analysis, create BTD fragments, or fragments of SAs, given prior use in similar projects, contexts, or CONOPS. More generally, we would like to be able to *synthesize* candidate BTDs, and thereby SAs, given a library of domain knowledge consisting of possible controls, barriers, their dependencies, orderings, recovery mechanisms, specification of functional capabilities, e.g., in terms of the threats mitigated, or measures such as effectiveness, integrity, etc. Moreover such reuse should be carefully executed to account for the inevitable differences in usage contexts.

As such, we envision extending the methodology and toolset to be applicable for use in other safety-critical application domains besides aviation.

Acknowledgement

This work was supported by the Safe Autonomous Systems Operations (SASO) project, under the Airspace Operations and Safety Program (AOSP) of the Aeronautics Research Mission Directorate (ARMD) at the National Aeronautics and Space Administration (NASA).

References

[1] E. Denney, G. Pai, I. Whiteside, Modeling the Safety Architecture of UAS Flight Operations, in: S. Tonetta, E. Schoitsch, F. Bitsch (Eds.), Computer Safety, Reliability, and Security. SAFECOMP 2017., Vol. 10488 of Lecture Notes in Computer Science, Springer, Cham, 2017. doi:10.1007/978-3-319-66266-4_11.

[2] FAA Air Traffic Organization, Safety and Technical Training Service Unit, Transforming Risk Management: Understanding the Challenges of Safety Risk Measurement, https://go.usa.gov/xXxea (Dec. 2016).

[3] UK Civil Aviation Authority (CAA), Bowtie risk assessment models, http://www.caa.co.uk/Safety-Initiatives-and-Resources/Working-with-industry/Bowtie/ (2015).

[4] A. P. Acfield, R. A. Weaver, Integrating safety management through the bowtie concept: A move away from the safety case focus, in: Proceedings of the Australian System Safety Conference (ASSC 2012), Vol. 145, CRPIT, 2012, pp. 3–12.

[5] R. A. Clothier, B. P. Williams, N. L. Fulton, Structuring the safety case for unmanned aircraft system operations in non-segregated airspace, Safety Science 79 (2015) 213 – 228. doi:10.1016/j.ssci.2015.06.007.

[6] Joint Authorities for Rulemaking of Unmanned Systems (JARUS), JARUS guidelines on Specific Operations Risk Assessment (SORA), Final/Public Release Ed. 1.0 (Jun. 2017).
URL http://jarus-rpas.org/content/jar-doc-06-sora-package

[7] E. Denney, G. Pai, Safety considerations for UAS ground-based detect and avoid, in: 2016 IEEE/AIAA 35th Digital Avionics Systems Conference (DASC), 2016, pp. 1–10. doi:10.1109/DASC.2016.7778077.

[8] T. Prevot, J. Rios, P. Kopardekar, J. Robinson III, M. Johnson, J. Jung, UAS Traffic Management (UTM) Concept of Operations to Safely Enable Low Altitude Flight Operations, in: Proceedings of 16th AIAA Aviation Technology, Integration, and Operations Conference, no. AIAA-2016-3292, 2016. doi:10.2514/6.2016-3292.

[9] E. Denney, G. Pai, I. Whiteside, Model-driven development of safety architectures, in: 2017 ACM/IEEE 20th International Conference on Model Driven Engineering Languages and Systems (MODELS), 2017, pp. 156–166. doi:10.1109/MODELS.2017.27.

[10] US Department of Transportation, Federal Aviation Administration, Safety Risk Management Policy, National Policy, Order 8404.4B (May 2017).
URL https://www.faa.gov/documentLibrary/media/Order/FAA_Order_8040.4B.pdf

[11] FAA Air Traffic Organization, Safety Management System Manual version 4.0, Federal Aviation Administration (May 2014).
URL https://www.faa.gov/air_traffic/publications/media/ATO-SMS-Manual.pdf

[12] E. Denney, G. Pai, Tool support for assurance case development, Journal of Automated Software Engineering 25 (3) (2018) 435–499. doi:10.1007/s10515-017-0230-5.

[13] Adelard LLP, Assurance and Safety Case Environment (ASCE), http://www.adelard.com/asce/ (2011).

[14] E. Denney, G. Pai, A methodology for the development of assurance arguments for unmanned aircraft systems, in: Proceedings of the 33rd International System Safety Conference (ISSC), 2015.

[15] R. Clothier, E. Denney, G. Pai, Making a Risk Informed Safety Case for Small Unmanned Aircraft System Operations, in: Proceedings of the 17th AIAA Aviation Technology, Integration, and Operations Conference (ATIO 2017), AIAA Aviation Forum, no. (AIAA 2017-3275), 2017. doi:10.2514/6.2017-3275.

[16] E. Denney, G. Pai, Argument-based airworthiness assurance of small UAS, in: Proceedings of the 34th IEEE/AIAA Digital Avionics Systems Conference (DASC), 2015, pp. 5E4–1–5E4–17. doi:10.1109/DASC.2015.7311439.

[17] E. Denney, G. Pai, Architecting a Safety Case for UAS Flight Operations, in: 34th International System Safety Conference (ISSC), 2016.

[18] International Electrotechnical Commission (IEC), Functional Safety of Electrical/Electronic/Programmable Electronic Safety-related Systems, IEC 61508 (2010).

[19] P. Feiler, D. Gluch, J. Mcgregor, An architecture-led safety analysis method, in: Proceedings of the 8th European Congress on Embedded Real Time Software and Systems (ERTS 2016), 2016.

[20] J. Dugan, G. Pai, H. Xu, Combining Software Quality Analysis with Dynamic Event/Fault Trees for High Assurance Systems Engineering, in: Proceedings of the 10th IEEE High Assurance Systems Engineering Conference (HASE), 2007, pp. 245– 255. doi:10.1109/HASE.2007.73.

[21] The Assurance Case Working Group (ACWG), Goal Structuring Notation Community Standard Version 2 (Jan. 2018).
URL https://scsc.uk/r141B:1

[22] UK Ministry of Defence (MOD), Safety management requirements for defence systems, Defence Standard 00-56, Issue 7 (2017).

[23] US Department of Transportation, Federal Aviation Administration (FAA), Flight Standards Information Management System, Volume 16, Unmanned Aircraft Systems, Order 8900.1 (Jun. 2014).

[24] European Organisation for the Safety of Air Navigation (EUROCONTROL), Safety Case Development Manual, 2nd Edition, no. DAP/SSH/091, 2006.

[25] UK Civil Aviation Authority (CAA) Safety and Airspace Regulation Group, Unmanned Aircraft System Operations in UK Airspace – Guidance, CAP722, 6th Ed. (March 2015).

[26] International Civil Aviation Organization (ICAO) Asia and Pacific Office, Building a Safety Case for Delivery of an ADS-B Separation Service, Guidance Material v1.0 (Sep. 2011).

[27] NASA Office of Safety and Mission Assurance, NASA General Safety Program Requirements, NPR 8715.3D (Aug. 2017).
URL https://nodis3.gsfc.nasa.gov/displayDir.cfm?Internal_ID=N_PR_8715_003D_

[28] Office of Safety and Mission Assurance, NPR 8715.5B, Range Flight Safety Program, NASA (Feb. 2018).
URL https://nodis3.gsfc.nasa.gov/displayDir.cfm?t=NPR&c=8715&s=5A

[29] Joint Authorities for Rulemaking of Unmanned Systems (JARUS), JARUS guidelines on Specific Operations Risk Assessment (SORA) (External Consultation Draft) (Aug. 2016).

[30] S-18, Aircraft And System Development And Safety Assessment Committee, ARP 4761, Guidelines and Methods for Conducting the Safety Assessment Process on Civil Airborne Systems and Equipment, Society of Automotive Engineers (SAE) (Dec. 1996).

[31] N. J. Duijm, Safety-barrier diagrams as a safety management tool, Reliability Engineering and System Safety 94 (2) (2009) 332–341. doi:10.1016/j.ress.2008.03.031.
[32] E. Ruijters, M. Stoelinga, Fault tree analysis: A survey of the state-of-the-art in modeling, analysis and tools, Computer Science Review 15-16 (2015) 29 – 62. doi:10.1016/j.cosrev.2015.03.001.
[33] E. Denney, G. Pai, Composition of safety argument patterns, in: A. Skavhaug, J. Guiochet, F. Bitsch (Eds.), Computer Safety, Reliability and Security. SAFECOMP 2016., Vol. 9922 of Lecture Notes in Computer Science, Springer, Cham, 2016. doi:10.1007/978-3-319-45477-1_5.
[34] L. Xing, J. B. Dugan, Analysis of generalized phased-mission system reliability, performance, and sensitivity, IEEE Transactions on Reliability 51 (2) (2002) 199–211. doi:10.1109/TR.2002.1011526.
[35] A. Rae, R. Alexander, J. McDermid, Fixing the cracks in the crystal ball: A maturity model for quantitative risk assessment, Reliability Engineering and System Safety 125 (2014) 67 – 81. doi:https://doi.org/10.1016/j.ress.2013.09.008.
[36] P. Brooker, Air Traffic Management accident risk. Part 1: The limits of realistic modelling, Safety Science 44 (5) (2006) 419–450. doi:10.1016/j.ssci.2005.11.004.
[37] P. Bishop, R. Bloomfield, B. Littlewood, A. Povyakalo, D. Wright, Towards a formalism for conservative claims about the dependability of software-based systems, IEEE Transactions on Software Engineering 37 (5) (2011) 708–717. doi:10.1109/TSE.2010.67.
[38] D. Steinberg, F. Budinsky, M. Paternostro, E. Merks, EMF: Eclipse Modeling Framework 2.0, 2nd Edition, Addison-Wesley Professional, 2009.
[39] E. Denney, I. Habli, G. Pai, Dynamic safety cases for through-life safety assurance, in: Proceedings of the 37th International Conference on Software Engineering (ICSE 2015): New Ideas and Emerging Results track (NIER), Florence, Italy, 2015. doi:10.1109/ICSE.2015.199.
[40] E. Denney, D. Naylor, G. Pai, Querying Safety Cases, in: A. Bondavalli, F. D. Giandomenico (Eds.), Computer Safety, Reliability and Security. SAFECOMP 2014., Vol. 8666 of Lecture Notes in Computer Science, Springer, 2014, pp. 294–309. doi:10.1007/978-3-319-10506-2_20.
[41] J. B. Dugan, Automated analysis of phased-mission reliability, IEEE Transactions on Reliability 40 (1) (1991) 45–52, 55. doi:10.1109/24.75332.
[42] E. Denney, M. Johnson, G. Pai, Towards a Rigorous Basis for Specific Operations Risk Assessment of UAS, in: Proceedings of the 37th AIAA/IEEE Digital Avionics Systems Conference (DASC 2018), 2018.