

Modeling Safety Case Evolution - Examples from the Air Traffic Management Domain

Massimo Felici

LFCS, School of Informatics, The University of Edinburgh
Edinburgh EH9 3JZ, UK
mfelici@inf.ed.ac.uk
<http://homepages.inf.ed.ac.uk/mfelici/>

Abstract. In order realistically and cost-effectively to realize the ATM (Air Traffic Management) 2000+ Strategy, systems from different suppliers will be interconnected to form a complete functional and operational environment, covering ground segments and aerospace. Industry will be involved as early as possible in the lifecycle of ATM projects. EUROCONTROL manages the processes that involve the definition and validation of new ATM solutions using Industry capabilities (e.g., SMEs). In practice, safety analyses adapt and reuse system design models (produced by third parties). Technical, organisational and cost-related reasons often determine this choice, although design models are unfit for safety analysis. This paper is concerned with evolutionary aspects in judging safety for ATM systems. The main objective is to highlight a model specifically targeted to support evolutionary safety analysis. The systematic production of safety analysis (models) will decrease the cost of conducting safety analysis by supporting reuse in future ATM projects.

1 Introduction

The future development of Air Traffic Management (ATM), set by the ATM 2000+ Strategy [11], involves a structural revision of ATM processes, a new ATM concept and a systems approach for the ATM network. The overall objective [11] is, *for all phases of flight, to enable the safe, economic, expeditious and orderly flow of traffic through the provision of ATM services, which are adaptable and scalable to the requirements of all users and areas of European airspace.* This requires ATM services to go through significant structural, operational and cultural changes that will contribute towards the ATM 2000+ Strategy. Moreover, from a technology viewpoint, future ATM services will employ new systems forming the emergent ATM architecture underlying and supporting the European Commission's Single European Sky Initiative.

ATM services, it is foreseen, will need to accommodate an increasing traffic, as many as twice the number of flights, by 2020. This challenging target will require the cost-effective gaining of extra capacity together with the increase of safety levels [31, 32]. Enhancing safety levels affects the ability to accommodate increased traffic demand as well as the operational efficiency of ensuring safe

separation between aircraft. Suitable safety conditions shall precede the achievement of increased capacity (in terms of accommodated flights). Therefore, it is necessary to foresee and mitigate safety issues in aviation where ATM can potentially deliver safety improvements.

In particular, there are *complex interactions* [34] between aircrafts and ATM safety functions. Unfortunately, these complex interactions may give rise to catastrophic failures. For instance, the accident (1 July 2002) between a BOEING B757-200 and a Tupolev TU154M [5], that caused the fatal injuries of 71 persons, provides an instance of unforeseen complex interactions. These interactions triggered a catastrophic failure, although all aircraft systems were functioning properly [5]. Humans [17, 33] using complex languages and procedures mediate these interactions. It is necessary further to understand how humans use external artifacts (e.g., tools) to mediate complex interactions. This would allow the understanding of how humans adopt technological artifacts and adapt their behaviours in order to accommodate ATM technological evolution. Unfortunately, the evolution of technological systems often corresponds to a decrease in technology trust affecting work practice [8]. Work practice and systems evolve rapidly in response to demand and a culture of continuous improvements [34]. A comprehensive account of ATM systems would allow the modeling of evolution. This will enhance strategies for deploying new system configurations or major system upgrades. On the one hand, modeling and understanding system evolution support the engineering of (evolving) ATM systems. On the other hand, modeling and understanding system evolution allow the communication of changes across different organisational levels [37]. This would enhance visibility of system evolution as well as trust in transition to operations.

Introducing safety relevant systems in ATM contexts requires us to understand the risk involved in order to mitigate the impact of possible failures. Safety analysis involves the activities (i.e., system identification and definition, risk analysis in terms of tolerable severity and frequency of hazards, definition of mitigation actions) that allow the systematic identification of hazards, risk assessment and mitigation processes in safety-critical systems [28, 41]. These general activities are deemed acceptable in diverse safety-critical domains (e.g., nuclear and chemical plants), which allow the unproblematic application of conventional safety analysis [28, 41]. Some safety-critical systems (e.g., nuclear or chemical plants) are well-confined entities with limited predictable interactions with the surroundings. Physical design structures constrain system interactions and stress the separation of safety related components from other system parts. This ensures the independence of failures. Therefore, in some safety-critical domains it is possible to identify acceptable tradeoffs between completeness and manageability during the definition and identification of the system under analysis. In contrast, ATM systems operate in open and dynamic environments. Hence, it is difficult to identify the full picture of system interactions in ATM contexts.

Recent safety requirements, defined by EUROCONTROL (European organization for the safety of air navigation), imply the adoption of a similar safety analysis for the introduction of new systems and their related procedures in the

ATM domain [10]. Unfortunately, ATM systems and procedures have distinct characteristics (e.g., openness, volatility, etc.) that expose limitations of the approach. In particular, the complete identification of the system under analysis is crucial for its influence on the cost and the effectiveness of the safety analysis. Therefore, safety analysis has to take into account complex interaction mechanisms (e.g., failure dependence, reliance in ATM, etc.) in order to guarantee and even increase the overall ATM safety as envisaged by the ATM 2000+ Strategy.

This paper is concerned with limitations of safety analysis with respect to evolution. The paper is structured as follows. Section 2 describes safety analysis in the ATM domain. Section 3 proposes a framework that enhances evolutionary safety analysis. Section 4 uses the ASCETM tool for the analysis of safety case changes. Section 5 introduces a logical framework for modeling and capturing safety case changes. The framework enhances the understanding of safety case evolution. Finally, Section 6 draws some conclusions.

2 Safety Analysis in ATM

ATM services across Europe are constantly changing in order to fulfil the requirements identified by the ATM 2000+ Strategy [11]. Currently, ATM services are going through a structural revision of processes, systems and underlying ATM concepts. This highlights a systems approach for the ATM network. The delivery and deployment of new systems will let a new ATM architecture emerge. The EUROCONTROL OATA project [38] intends to deliver the concepts of operation, the logical architecture in the form of a description of the interoperable system modules, and the architecture evolution plan. All this will form the basis for common European regulations as part of the *Single European Sky*.

The increasing integration, automation and complexity of ATM systems requires a systematic and structured approach to risk assessment and mitigation, including hazard identification, as well as the use of predictive and monitoring techniques to assist in these processes. Faults [27] in the design, operation or maintenance of ATM systems or errors in ATM systems could affect the safety margins (e.g., loss of separation) and result in, or contribute to, an increased hazard to aircrafts or a failure (e.g., a loss of separation and an accident in the worst case). Increasingly, ATM systems rely on the reliance (e.g., the ability to recover from failures and accommodate errors) and safety (e.g., the ability to guarantee failure independence) features placed upon all system parts. Moreover, the increased interaction of ATM across state boundaries requires that a consistent and more structured approach be taken to the risk assessment and mitigation of all ATM System elements throughout the ECAC (European Civil Aviation Conference) states [9]. Although the average trends show a decrease in the number of fatal accidents for Europe, the approach and landing accidents are still the most safety pressing problems facing the aviation industry [35, 36, 42]. Unfortunately, even maintaining the same safety levels across the European airspace would be insufficient to accommodate an increasing traffic without affecting the overall safety of ATM Systems [7].

The introduction of new safety relevant systems in ATM contexts requires us to understand the risk involved in order to mitigate the impact of possible failures. The EUROCONTROL Safety Regulatory Requirement [10], ESARR4, requires the use of a risk based-approach in ATM when introducing and/or planning changes to any (ground as well as onboard) part of ATM systems. This concerns the human, procedural and equipment (i.e., hardware or software) elements of ATM systems as well as its environment of operations at any stage of the life cycle of the ATM System. The ESARR4 [10] requires that ATM service providers systematically identify any hazard for any change into ATM systems. Moreover, they have to assess any related risk and identify relevant mitigation actions. In order to provide guidelines for and standardise safety analysis EUROCONTROL has developed the EATMP Safety Assessment Methodology (SAM) [12] reflecting best practices for safety assessment of Air Navigation Systems.

The SAM methodology provides a means of compliance to ESARR4. The objective of the methodology is to define the means for providing assurance that an Air Navigation System is safe for operational use. The SAM methodology describes a generic process for the safety assessment of Air Navigation Systems. This process consists of three major steps: *Functional Hazard Assessment (FHA)*, *Preliminary System Safety Assessment (PSSA)* and *System Safety Assessment (SSA)*. The process covers the complete lifecycle of an Air Navigation System, from initial system definition, through design, implementation, integration, transfer to operations and maintenance. Although the SAM methodology describes the underlying principles of the safety assessment process, it provides limited information how to apply these principles in specific projects.

3 Evolutionary Safety Analysis

Evolutionary Safety Analysis [15] relies on a logical framework that captures cycles of discoveries and exploitations. The underlying idea involves the identification of mappings between socio-technical solutions and problems. The proposed framework [15] exploits these mappings in order to construct an evolutionary model that enhances safety analysis. This section briefly recalls the proposed framework for evolutionary safety analysis. The remainder of the paper shows a particular aspect of the framework, which captures safety case evolution. The examples drawn from the ATM domain emphasise how evolutionary safety analysis supports work practice. Figure 1 shows the framework for evolutionary safety analysis. The framework captures these evolutionary cycles at different levels of abstraction and on diverse models. This paper explicitly develops the evolution of safety cases. The framework consists of three different hierarchical layers: *System Modeling Transformation (SMT)*, *Safety Analysis Modeling Transformation (SAMT)* and *Operational Modeling Transformation (OMT)*. The remainder of this section describes the three hierarchical layers.

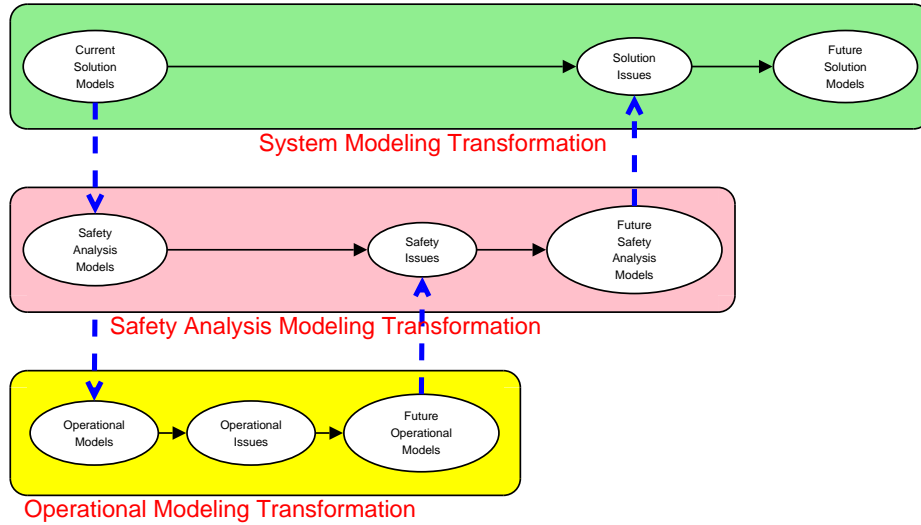


Fig. 1. A framework for modeling evolutionary safety analyses.

3.1 System Modeling Transformation

The definition and identification of the system under analysis is extremely critical in the ATM domain. System models defined during the design phase are adapted and reused for safety and risk analysis. Organizational and cost-related reasons often determine this choice, without questioning whether models are suitable for the intended use. System models capture characteristics that may be of primary importance for design, but irrelevant for safety analysis. The main drawback is that design models are tailored to support the work of system designers. Models should be working-tools that, depending on their intended use, ease and support specific activities and cognitive operations of users.

Heterogeneous engineering [29] provides a comprehensive viewpoint, which allows us to understand the underlying mechanisms of evolution of socio-technical systems. Heterogeneous engineering involves both the systems approach [22] as well as the social shaping of technology [30]. According to heterogeneous engineering, system requirements specify mappings between problem and solution spaces. Both spaces are socially constructed and negotiated through sequences of mappings between solution spaces and problem spaces [3, 4]. Therefore, system requirements emerge as a set of consecutive solution spaces justified by a problem space of concerns to stakeholders. Requirements, as mappings between socio-technical solutions and problems, represent an account of the history of socio-technical issues arising and being solved within industrial settings. The formal extension of these mappings (or solution space transformations) identifies a framework to model and capture evolutionary system features (e.g., requirements evolution, evolutionary dependencies, etc.) [14].

System Modeling Transformation captures how solution models evolve in order to accommodate design issues or evolving requirements. Therefore, an SMT captures system requirements as mappings between socio-technical solutions and problems. This allows the gathering of changes into design solutions. That is, it is possible to identify how changes affect design solution. Moreover, this enables sensitivity analyses of design changes. In particular, this allows the revision of safety requirements and the identification of hazards due to the introduction of a new system. Therefore, the SMT supports the gathering of safety requirements for evolving systems. That is, it supports the main activities occurring during the top-down iterative process FHA in the SAM methodology [12]. The FHA in the SAM methodology then initiates another top-down iterative approach, i.e., the PSSA. Similarly, the framework considers design solutions and safety objectives as input to Safety Analysis. Safety analysis assesses whether the proposed design solution satisfies the identified safety objectives. This phase involves different methodologies (e.g., Fault Tree Analysis, HAZOP, etc.) that produce diverse (system) models. System usage or operational trials may give rise to unforeseen safety issues that invalidate (parts of) safety models. In order to take into account these issues, it is necessary to modify safety analysis. Therefore, safety analysis models evolve too.

3.2 Safety Analysis Modeling Transformation

The failure of safety-critical systems highlights safety issues [23, 28, 34, 41]. It is often the case that diverse causes interacted and triggered particular unsafe conditions. Although safety analysis (i.e., safety case) argues system safety, complex interactions, giving rise to failures, expose the limits of safety arguments. Therefore, it is necessary to take changes into account in safety arguments [18]. Greenwell, Strunk and Knight in [18] propose an enhanced safety-case lifecycle by evolutionary (safety-case) examples drawn from the aviation domain. The lifecycle identifies a general process for the revision of safety cases.

Figures 2 and 3 show subsequent versions of a safety case. The graphical notation that represents the safety cases is the Goal Structuring Notation (GSN) [24]. Although GSN addresses the maintenance of safety cases, the approach provides limited support with respect to complex dependencies (e.g., external to the safety argument) [25]. Moreover, it lacks any interpretation of the relationships between subsequent safety cases. Figure 2 shows the initial safety case arguing: “*Controller aware of altitude violations*”. Unfortunately, an accident invalidates the justification J1. The satisfaction of the subgoal G2 is insufficient for the satisfaction of the goal G1. Figure 3 shows the revised safety case that addresses the issue occurred. Unfortunately, another accident, again, invalidates the second safety case [18]. Hence, the safety argument needs further revision in order to address the safety flaw uncovered by the accident. Safety Analysis Modeling Transformation, similarly to the SMT, captures how safety analysis models evolve in order to accommodate emerging safety issues. Although design models serve as a basis for safety models, they provide limited support to capture unforeseen system interactions. Therefore, SAMT supports those activities involved in

the PSSA process of the SAM methodology [12]. Note that although the SAM methodology stresses that both FHA and PSSA are iterative processes, it provides little support to manage process iterations as well as system evolution in terms of design solution and safety requirements. The framework supports these evolutionary processes.

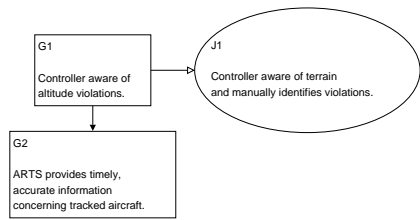


Fig. 2. Initial safety argument.

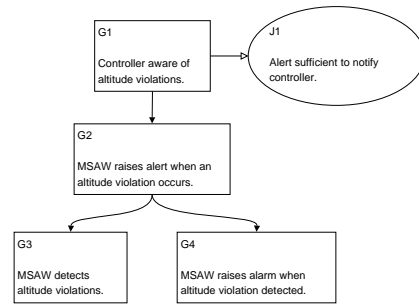


Fig. 3. Revised safety argument.

3.3 Operational Modeling Transformation

Operational models (e.g., structured scenarios, patterns of interactions, structured procedures, workflows, etc.) capture heterogeneous system dynamics. Unfortunately, operational profiles often change with system usage (in order to integrate different functionalities or to accommodate system failures). The main problem areas (e.g., ATC Human Performance, Flight Crew Human Performance, Aircraft, etc.) identified in Controller Reports [1] and TCAS II Incidents [2] highlight the complexity and the coupling within the ATM domain [34]. The analysis [15] of the reports is in agreement with other studies [39, 43] that analyse human errors as organizational failures [20, 28, 37].

Capturing operational interactions and procedures allows the analysis of human reliability [20]. In a continuously changing environment like ATM, adaption enhances the coupling between man and machine [21]. Hollnagel in [21] identifies three different adaption strategies: *Adaption Through Design*, *Adaption through Performance* and *Adaption through Management*. Operational Modeling Transformation captures how operational models change in order to accommodate issues arising. The evolution of operation models informs safety analyses of new hazards. Therefore, OMT supports the activities involved in the SSA process of the SAM methodology.

4 Safety Case Changes

This section uses the (Assurance and Safety Case Environment) ASCE™ tool (see Appendix A) for the analysis of safety case changes. The ASCE Difference

Tool v1.1 supports the analysis of differences between two ASCE networks (i.e., safety cases). Note that this functionality has been recently released with ASCE™ v3.0. Although it is possible to analyse safety case changes in small and simple examples, the tool supports the automation of safety case management and analysis. This further stresses how capturing safety case evolution would support safety case judgement and safety analysis practice. Figure 4 shows a report from the ASCE™ Difference Tool.

Node differences

Please note: When analysing changes in Node narratives (HTML Content), only new and deleted sentences are shown. Formatting changes, moved sentences and copied/duplicated sentences are **not shown**.

Node N7737803 (N7737803 - J1) was changed

Node HTML narrative content was changed/edited.

Removed sentences	Inserted sentences
Controller aware of terrain and manually identifies violations. (approx 0% into document)	Alert sufficient to notify controller. (approx 0% into document)

Node N1813777 (N1813777 - G2) was changed

Node HTML narrative content was changed/edited.

Removed sentences	Inserted sentences
ARTS provides timely, accurate information concerning tracked aircraft. (approx 0% into document)	MSAW raises alert when an altitude violation occurs. (approx 0% into document)

Supporting Link (type = [1]) from Node [N7822475 - G3] was added.
Supporting Link (type = [1]) from Node [N8116556 - G4] was added.

Node N7822475 (N7822475 - G3) was added.

Node N8116556 (N8116556 - G4) was added.

Fig. 4. Analysing differences between two ASCE networks.

The report consists of the comparison of the two subsequent safety cases, i.e., Figure 2 and Figure 3. The analysis points out the differences (i.e., safety case changes) between the subsequent safety cases. It clearly points out the safety case changes: changing the nodes J1 and G2; adding the two new nodes G3 and G4. The ASCE™ Difference Tool detects and reports on structural and content differences between two networks, such as: General configuration changes (e.g., project name, version, author, description); Added new and deleted nodes; Modified node attributes (e.g., type, id, title, status fields), Modified node content, and optionally new and deleted sentences in the node narratives; Structural changes (i.e. new, deleted and modified links).

The identification of differences between two ASCE networks relies on the comparison between safety case structures. This further highlights the impor-

tance of comparing and representing structured safety cases. Related research [26] highlights the role of structured safety cases in the analysis of the whole ATM airspace. The findings of comparing different structured safety cases [26] stress the importance of understanding how structured safety cases change [26]. Hence, capturing safety case changes would support safety case judgement and safety analysis.

5 Modeling Safety Case Changes

This section shows that it is possible to capture safety case changes in a logical framework. The logical framework is similar to the one underlying System Modeling Transformations (SMTs) for modeling requirements evolution [14]. Requirements, as mappings between socio-technical solutions and problems, represent an account of the history of socio-technical issues arising and being solved within industrial settings [3, 4]. These mappings identify a *Functional Ecology* model that defines requirements as emerging from solution space transformations. The Functional Ecology model describes solution-problem iterations of the form: *solution* \rightarrow *problem* \rightarrow *solution*. A solution space solves some highlighted problems. The contextualization of the selected problems into the initial solution space identifies the system requirements as mappings between solution and problem spaces. The resolution of these problems identifies a future solution. The mappings between the solved problems and the future solution define further system requirements. This heterogeneous account of requirements is convenient to capture requirements evolution [14]. This implies that requirements engineering processes consist of solutions searching for problems, rather than the other way around (that is, problems searching for solutions) [3, 4].

This section extends the use of evolutionary modeling [14] to safety case evolution, hence *safety space transformation*. Modeling safety case changes relies on a formal extension of solution space transformations [14]. The basic idea is to provide a formal representation of solutions (i.e., safety cases) and problems. The aim of a formal representation is twofold. On the one hand the formalisation of safety cases and problems supports *model-driven judgement*. On the other hand it allows us to formally capture *safety space transformations*, hence *safety case evolution*. The formalisation represents safety cases and problems in terms of modal logic [6, 16]. Propositional modal logic provides enough expressiveness in order to formalise safety space transformations. The formal representation relies on logic bases: syntax, semantics and proof systems¹. All definitions can be naturally extended in terms of other logics (e.g., [13, 40]) bases (i.e., syntax, semantics and proof systems). The definitions still remain sound and valid due to construction arguments.

¹ Note that there exist different logics (e.g., **K**, **D**, **K4**, **S4**, **S5**, etc.) that correspond to different proof systems [6, 16]. Any specific proof system implies particular features to the models that can be proved. It is beyond the scope of this work to decide which proof system should be used in any specific case.

Intuitively, a *Safety Space* is just a collection of safety cases, which represent the organisational knowledge acquired by the social shaping of technical systems. The *Current Safety Space*, denoted as \mathcal{S}_t , embodies the history of solved social, technical, economic and procedural problems (i.e., socio-technical problems) that constitute the legacy of previously solved organisational problems at current time t . The Current Safety Space exists within a *Local Safety Space*. That is, \mathcal{S}_t is a subspace of \mathcal{S} . The Current Safety Space therefore captures the knowledge acquired by organisational learning (i.e., the previously solved organisational problems). In other words, the Current Safety Space consists of the adopted solutions due to organisational learning. This definition further supports the assumption that safety space transformations capture organisational learning and system safety judgement, hence safety case evolution. Safety goals therefore are *accessible possibilities* or *possible worlds* in safety spaces available in the production environment. This intentionally recalls the notion of possible world underlying *Kripke models* [13]. Thus, safety cases are Kripke models. It is moreover possible to model the Current Safety Space in terms of Kripke models. \mathcal{S}_t is a collection of Kripke models. Note that Kripke models enable reasoning about knowledge [13] and uncertainties [19].

Let us briefly recall the notion of a Kripke model. A Kripke model, \mathcal{M} , consists of a collection G of *possible worlds*, an *accessibility relation* R on possible worlds and a mapping \Vdash between possible worlds and propositional letters. The \Vdash relation defines which propositional letters are true at which possible worlds. Thus, \mathcal{S}_t is a collection of countable elements of the form

$$\mathcal{M}_i^t = \langle G_i^t, R_i^t, \Vdash_i^t \rangle . \quad (1)$$

Each Kripke model then represents an available safety case. Thus, a Kripke model is a system of worlds in which each world has some (possibly empty) set of alternatives. The accessibility relation (or alternativeness relation), denoted by R , so that $\Gamma R \Delta$ means that Δ is an alternative (or possible) world for Γ . For every world Γ , an atomic proposition is either true or false in it and the truth-values of compound non-modal propositions are determined by the usual truth-tables. A modal proposition $\Box\varphi$ is regarded to be true in a world Γ , if φ is true in all the worlds accessible from Γ . Whereas, $\Diamond\varphi$ is true in Γ , if φ is true at least in one world Δ such that $\Gamma R \Delta$. In general, many safety cases may address a given problem. The resolution of various problems, hence the acquisition of further knowledge, narrows the safety space by refining the available safety cases.

Problems are formulae of (propositional) modal logic. Collections of problems (i.e., problem spaces) are issues (or believed so) arising during system production. Kripke models (i.e., solutions) provide the semantics in order to interpret the validity of (propositional) modalities (i.e., problems). Note that it is possible to adopt different semantics of the accessibility relations in Kripke models. For instance, the accessibility relation can capture information like safety case dependencies (or dependencies between safety goals). Using different semantics for interpreting the accessibility relation highlights and captures diverse evolutionary information. On the one hand, the use of different semantics highlights

the flexibility of the given framework. On the other hand, it requires careful considerations when used in practice to capture diverse evolutionary aspects of safety cases. Based on the syntax of Kripke models, proof systems (e.g., *Tableau systems*) consist of procedural rules (i.e., inference rules) that allow us to prove formula validity or to find counterexamples (or countermodels).

The mappings between the Current Safety Space \mathcal{S}_t and the *Proposed Safety Problem Space* \mathcal{P}_t identify safety requirements (demands, needs or desires of stakeholders) that correspond to problems as contextualised by a current safety case. These mappings represent the *safety arguments*. Let \mathcal{S}_t be the Current Safety Space and \mathcal{P}_t be the Proposed Safety Problem Space. The safety arguments \mathbf{SC}_a^t consists of the mappings (i.e., pairs) that correspond to each problem P_j^t in \mathcal{P}_t contextualised by a safety case \mathcal{M}_i^t in \mathcal{S}_t . Thus, for any possible world Γ in a Kripke model $\mathcal{M}_i^t \in \mathcal{S}_t$ and for any problem $P_j^t \in \mathcal{P}_t$ such that $(\mathcal{M}_i^t, \Gamma) \not\models_i^t P_j^t$, the pair $\langle \Gamma, P_j^t \rangle$ belongs to \mathbf{SC}_a^t . In formula,

$$\mathbf{SC}_a^t = \{ \langle \Gamma, P_j^t \rangle \mid (\mathcal{M}_i^t, \Gamma) \not\models_i^t P_j^t \} . \quad (2)$$

The final step of the Safety Space Transformation consists of the reconciliation of the Safety Space \mathcal{S}_t with the Proposed Safety Problem Space \mathcal{P}_t into a Proposed Safety Space \mathcal{S}_{t+1} (i.e., a subspace of a *Future Safety Space* \mathcal{S}'). The Proposed Safety Space \mathcal{S}_{t+1} takes into account (or solves) the selected problems. The resolution of the selected problems identifies the proposed future safety cases.

The reconciliation of \mathcal{S}_t with \mathcal{P}_t involves the resolution of the problems in \mathcal{P}_t . In logic terms, this means that the proposed solutions should satisfy the selected problems (or some of them). Note that the selected problems could be unsatisfiable as a whole (that is, any model is unable to satisfy all the formulae). This requires stakeholders to compromise (i.e., prioritise and refine) over the selected problems. The underlying logic framework allows us to identify model schemes that satisfy the selected problems. This requires to prove the validity of formulae by a proof system. If a formula is satisfiable (that is, there exist a model in which the formula is valid), it would be possible to derive by the proof system a model (or counterexample) that satisfies the formula. The reconciliation finally forces the identified model schemes into future solutions.

The final step of the Safety Space Transformation identifies mappings between the Proposed Safety Problem Space \mathcal{P}_t and the Proposed Safety Space \mathcal{S}_{t+1} . These mappings of problems looking for solutions represent the *safety constraints*. Let \mathcal{S}_t be the Current Safety Space, \mathcal{P}_t be the Proposed Safety Problem Space and \mathcal{S}_{t+1} be a Proposed Safety Space in a Future Safety Space \mathcal{S}' . The safety constraints in \mathbf{SC}_c^t consist of the mappings (i.e., pairs) that correspond to each problem P_j^t in \mathcal{P}_t solved by a safety case \mathcal{M}_i^{t+1} in \mathcal{S}_{t+1} . Thus, for any $\langle \Gamma, P_j^t \rangle \in \mathbf{SC}_a^t$, and for any Kripke model $\mathcal{M}_i^{t+1} \in \mathcal{S}_{t+1}$ that solves the problem $P_j^t \in \mathcal{P}_t$, the pair $\langle P_j^t, \Gamma \rangle$ belongs to \mathbf{SC}_c^t . In formula,

$$\mathbf{SC}_c^t = \{ \langle P_j^t, \Gamma \rangle \mid (\Gamma, P_j^t) \in \mathbf{SC}_a^t \text{ and } (\mathcal{M}_i^{t+1}, \Gamma) \models_i^{t+1} P_j^t \} . \quad (3)$$

Figure 5 shows a safety space transformation. The safety case transformation captures the changes from the initial safety case \mathcal{M}_i^t (see Figure 2) to the revised safety case \mathcal{M}_i^{t+1} (see Figure 3).

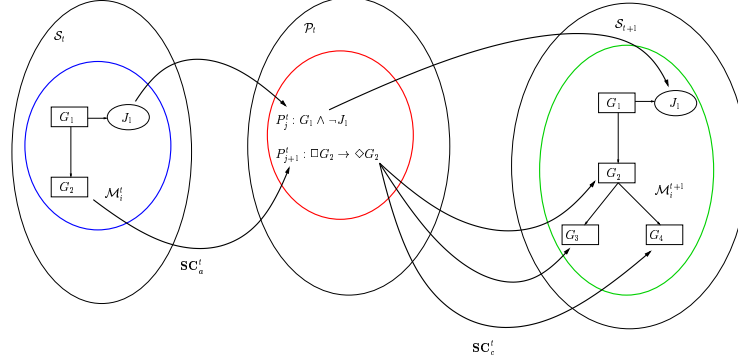


Fig. 5. A safety space transformation.

An accident invalidates the justification J1. The satisfaction of the subgoal G2 is insufficient for the satisfaction of the goal G1. The proposed safety problem space, \mathcal{P}_t , contains these problems, i.e., P_j^t and P_{j+1}^t . The safety space transformation addresses the highlighted problems into the proposed safety case \mathcal{M}_i^{t+1} . In order to address the highlighted problems, it is necessary to change the initial safety case. The proposed changes are taken into account in the proposed safety case. Note that there might be different proposed safety cases addressing the proposed safety problem space.

The safety space transformation identifies the safety case construction and judgement in terms of safety argumentations and constraints. The safety case consists of the collections of mappings between safety cases and problems. The first part of a safety case consists of the safety argumentations, which capture the relationship that comes from safety cases looking for problems. The second part of a safety case consists of the safety constraints, which capture how future safety cases address given problems. Safety cases at any given time, t , can be represented as the set of all the arcs, that reflect the contextualised connections between the problem space and the current and future safety space. In formula,

$$\mathbf{SC}^t = (\mathbf{SC}_a^t, \mathbf{SC}_c^t) \quad (4)$$

The definition of safety case transformation enables us further to interpret and understand safety case changes, hence safety case evolution [14].

6 Conclusions

This paper introduces a logical framework for modeling safety case evolution. The framework extends the use of evolutionary modeling [14] to *safety space*

transformation, hence *safety case evolution*. Modeling safety case changes relies on a formal extension of solution space transformations [14]. The underlying idea is to provide a formal representation of safety cases and problems. On the one hand, the formalisation of safety cases and problems supports *model-driven judgement*. On the other hand, it allows us to formally capture *safety space transformations*, hence *safety case evolution*. The modeling of safety case evolution provides new insights in safety case judgement and safety analysis.

Modeling Safety Case Evolution. The framework captures how safety cases evolve in order to accommodate arising safety problems. The safety space transformation identifies the safety case construction and judgement in terms of safety argumentations and constraints. The safety case consists of the collections of mappings between safety cases and problems. The first part of a safety case consists of the safety argumentations, which capture the relationship that comes from safety cases looking for problems. The second part of a safety case consists of the safety constraints, which capture how future safety cases address given problems. The definition of safety case transformation enables us further to interpret and understand safety case changes, hence safety case evolution. Therefore, the framework enables the implementation of evolutionary safety analysis [15].

Formal Tool Extensions. The framework provides a formal extension of the ASCE Difference Tool. The tool relies on the comparison between two structured safety cases. Therefore, the logical framework provides a formal support for interpreting and capturing safety case changes.

Support for Guidelines and Work Practice. The framework supports industry guidelines (e.g., SAM methodology [12]), which emphasise the iterative nature of safety analysis [12, 15]. Moreover, the framework supports safety judgement as well as evolutionary safety analysis [15]. The underlying evolutionary aspects characterise work practice in the safety analysis of continuously evolving industry domain (e.g., ATM).

Organisational Knowledge and Safety Judgement. The framework relies on basic logic models (i.e., Kripke models) that enable reasoning about knowledge [13] and uncertainty [19]. This highlights safety judgement (that is, the construction of safety cases) as an organisational process. That is, the safety judgement consists of gathering organisational knowledge about the system. This further highlights how organisational (knowledge) failures affect safety [28, 34, 37].

In conclusion, this paper introduces a framework for modeling safety case evolution. The framework supports safety judgement and evolutionary safety analysis. Future work intends to implement an ASCE plugin that supports safety case evolution. This would enhance and support the dissemination of safety case evolution practice among the safety case users.

Acknowledgments. I would like to thank George Cleland and Luke Emmet of Adelard - <http://www.adelard.com/> - for their help and comments on the As-

insurance and Safety Case Environment, ASCE™. ASCE™ is free for non commercial teaching and research purposes. ASCE™ V3.0 is available for download - <http://www.adelard.com/software/asce->. This work has been supported by the UK EPSRC Interdisciplinary Research Collaboration in Dependability, DIRC - <http://www.dirc.org.uk> - grant GR/N13999.

References

1. Aviation Safety Reporting System. *Controller Reports*, 2003.
2. Aviation Safety Reporting System. *TCAS II Incidents*, 2004.
3. M. Bergman, J. L. King, and K. Lyytinen. Large-scale requirements analysis as heterogeneous engineering. *Social Thinking - Software Practice*, pages 357–386, 2002.
4. M. Bergman, J. L. King, and K. Lyytinen. Large-scale requirements analysis revisited: The need for understanding the political ecology of requirements engineering. *Requirements Engineering*, 7(3):152–171, 2002.
5. BFU. *Investigation Report, AX001-1-2/02*, 2002.
6. A. Chagrov and M. Zakharyashev. *Modal Logic*. Number 35 in Oxford Logic Guides. Oxford University Press, 1997.
7. J. H. Enders, R. S. Dodd, and F. Fickeisen. Continuing airworthiness risk evaluation (CARE): An exploratory study. *Flight Safety Digest*, 18(9-10):1–51, September-October 1999.
8. EUROCONTROL. *Human Factor Module - Human Factors in the Development of Air Traffic Management Systems*, 1.0 edition, 1998.
9. EUROCONTROL. *EUROCONTROL Airspace Strategy for the ECAC States, ASM.ET1.ST03.4000-EAS-01-00*, 1.0 edition, 2001.
10. EUROCONTROL. *EUROCONTROL Safety Regulatory Requirements (ESARR). ESARR 4 - Risk Assessment and Mitigation in ATM*, 1.0 edition, 2001.
11. EUROCONTROL. *EUROCONTROL Air Traffic Management Strategy for the years 2000+*, 2003.
12. EUROCONTROL. *EUROCONTROL Air Navigation System Safety Assessment Methodology*, 2.0 edition, 2004.
13. R. Fagin, J. Y. Halpern, Y. Moses, and M. Y. Vardi. *Reasoning about Knowledge*. The MIT Press, 2003.
14. M. Felici. *Observational Models of Requirements Evolution*. PhD thesis, Laboratory for Foundations of Computer Science, School of Informatics, The University of Edinburgh, 2004.
15. M. Felici. Evolutionary safety analysis: Motivations from the air traffic management domain. In *Proceedings of the 24th International Conference on Computer Safety, Reliability and Security, SAFECOMP 2005*, 2005.
16. M. Fitting and R. L. Mendelsohn. *First-Order Modal Logic*. Kluwer Academic Publishers, 1998.
17. Flight Safety Fundation. *The Human Factors Implication for Flight Safety of Recent Developments In the Airline Industry*, number (22)3-4 in Flight Safety Digest, March-April 2003.
18. W. S. Greenwell, E. A. Strunk, and J. C. Knight. Failure analysis and the safety-case lifecycle. In *Proceedings of the IFIP Working Conference on Human Error, Safety and System Development (HESSD)*, pages 163–176, 2004.
19. J. Y. Halpern. *Reasoning about Uncertainty*. The MIT Press, 2003.

20. E. Hollnagel. *Human Reliability Analysis: Context and Control*. Academic Press, 1993.
21. E. Hollnagel. The art of efficient man-machine interaction: Improving the coupling between man and machine. In *Expertise and Technology: Cognition & Human-Computer Cooperation*, pages 229–241. Lawrence Erlbaum Associates, 1995.
22. A. C. Hughes and T. P. Hughes, editors. *Systems, Experts, and Computers: The Systems Approach in Management and Engineering, World War II and After*. The MIT Press, 2000.
23. C. W. Johnson. *Failure in Safety-Critical Systems: A Handbook of Accident and Incident Reporting*. University of Glasgow Press, Glasgow, Scotland, Oct. 2003.
24. T. P. Kelly. *Arguing Safety - A Systematic Approach to Managing Safety Cases*. PhD thesis, Department of Computer Science, University of York, 1998.
25. T. P. Kelly and J. A. McDermid. A systematic approach to safety case maintenance. In M. Felici, K. Kanoun, and A. Pasquini, editors, *Proceedings of the 18th International Conference on Computer Safety, Reliability and Security, SAFE-COMP'99*, number 1698 in LNCS, pages 13–26. Springer-Verlag, 1999.
26. S. Kinnersly. Whole airspace atm system safety case - preliminary study. Technical Report AEAT LD76008/2 Issue 1, AEA Technology, 2001.
27. J.-C. Laprie et al. Dependability handbook. Technical Report LAAS Report no 98-346, LIS LAAS-CNRS, Aug. 1998.
28. N. G. Leveson. *SAFWARE: System Safety and Computers*. Addison-Wesley, 1995.
29. D. A. MacKenzie. *Inventing Accuracy: A Historical Sociology of Nuclear Missile Guidance*. The MIT Press, 1990.
30. D. A. MacKenzie and J. Wajcman, editors. *The Social Shaping of Technology*. Open University Press, 2nd edition, 1999.
31. S. Matthews. Future developments and challenges in aviation safety. *Flight Safety Digest*, 21(11):1–12, Nov. 2002.
32. M. Overall. New pressures on aviation safety challenge safety management systems. *Flight Safety Digest*, 14(3):1–6, Mar. 1995.
33. A. Pasquini and S. Pozzi. Evaluation of air traffic management procedures - safety assessment in an experimental environment. *Reliability Engineering & System Safety*, 2004.
34. C. Perrow. *Normal Accidents: Living with High-Risk Technologies*. Princeton University Press, 1999.
35. H. Ranter. Airliner accident statistics 2002: Statistical summary of fatal multi-engine airliner accidents in 2002. Technical report, Aviation Safety Network, Jan. 2003.
36. H. Ranter. Airliner accident statistics 2003: Statistical summary of fatal multi-engine airliner accidents in 2003. Technical report, Aviation Safety Network, Jan. 2004.
37. J. Reason. *Managing the Risks of Organizational Accidents*. Ashgate Publishing Limited, 1997.
38. Review. Working towards a fully interoperable system: The EUROCONTROL overall ATM/CNS target architecture project (OATA). *Skyway*, 32:46–47, Spring 2004.
39. S. A. Shappell and D. A. Wiegmann. The human factors analysis and classification system - HFACS. Technical Report DOT/FAA/AM-00/7, FAA, Feb. 2000.
40. C. Stirling. *Modal and Temporal Properties of Processes*. Texts in Computer Science. Springer-Verlag, 2001.

41. N. Storey. *Safety-Critical Computer Systems*. Addison-Wesley, 1996.
42. G. W. van Es. A review of civil aviation accidents - air traffic management related accident: 1980-1999. In *Proceedings of the 4th International Air Traffic Management R&D Seminar*, New-Mexico, Dec. 2001.
43. D. A. Wiegmann and S. A. Shappell. A human error analysis of commercial aviation accidents using the human factors analysis and classification system (HFACS). Technical Report DOT/FAA/AM-01/3, FAA, Feb. 2001.

A ASCE™

The Assurance and Safety Case Environment, ASCE™, is a graphical hypertext tool which allows the development, review, analysis and dissemination of safety and assurance cases. It is based on the concept that a safety or assurance case should show the argument structure as well as providing evidence to support that argument. A safety case is “*a documented body of evidence that provides a demonstrable and valid argument that a system is adequately safe for a given application and environment over its lifetime*”. ASCE represents the underlying models/networks in XML (.axml files). This allows full programmatic access to the underlying content in order to implement any kind of required analysis. One major new functionality in ASCE™ V3.0 is the availability of *plugins*. Implementing analysis functionality as a plugin provides a way to analyse the safety case’s structure. Figure 6 shows an example application domain for ASCE™, i.e., the range of processes and activities that ASCE™ can support.

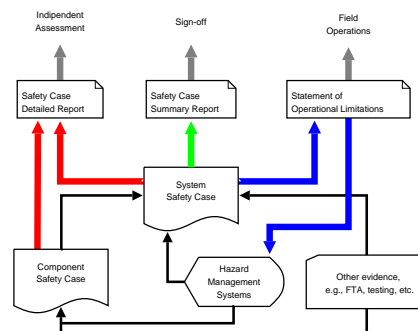


Fig. 6. An application domain for ASCE™.

The ASCE™ Difference Tool is a simple standalone tool for comparing two ASCE networks, and is very useful in supporting reviewing or auditing activities over some period whilst the ASCE network has been edited. It generates a report that summarises differences between any two ASCE networks.