

# Implementation of Change Management in Safety Cases

Mitesh Mistry    Massimo Felici\*  
School of Informatics, The University of Edinburgh

**This paper is concerned with change management for safety cases. Despite the necessity of maintaining and updating safety cases, change management practices have received little support yet. It is possible to identify overall maintenance processes or lifecycles for safety cases. However, these processes need to be supported in practice. This paper reports ongoing work about the implementation of a plugin supporting change management practices for safety cases. This enhances change management for safety cases. The resulting analysis (of the safety case evolution) would strengthen the safety case arguments and the confidence in them.**

*Keywords: Change Management, Safety Cases, Safety Case Maintenance, Safety Case Development, Safety Case Lifecycle.*

## 1. INTRODUCTION

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 [3][4][5]. A safety case consists of the following elements:

- Requirements: the safety objectives that must be met to guarantee safety
- Evidence: information from analysis and testing of the system in question
- Argument: showing how the evidence demonstrates compliance with the requirements
- Context: identifying the basis of the argument presented.

These four elements of a safety case are interdependent and in graphical notation are used to document and justify the safety claims for a particular system.

As systems evolve over time due to changing requirements, technologies and regulations, the safety case required to justify the safety of a system will need to be modified to maintain the strength and validity of the argument. This is a complex process in a safety critical domain and a means of facilitating this process is required, allowing for safety reviewers and system designers to better understand the evolution of safety cases; and thus through documenting these changes provide a stronger justification for the overall design and structure of the safety argument.

The purpose of this paper is to provide a practical design and implementation of a change management mechanism for Safety Cases. This paper will initially provide a background into safety cases, covering the basics for safety case construction using the GSN (Goal Structure Notation) [3][4], and their industrial application. The paper will then discuss and analyse the change management process in safety cases, focusing on the requirement for a means of managing and facilitating the process, and the potential impact that changes can have on the safety justification of the system. Forming the core of this project, the paper will describe a plug-in that is being developed for the ASCE (Assurance and Safety Case Environment)<sup>1</sup> tool, allowing for users to visually monitor the changes made to a safety case over time. The paper will then discuss the types of analysis that can be facilitated by the tool, concluding with a description of future tools that can be developed to aid the change management process.

## 2. SAFETY CASE MAINTENANCE

At present there are many notable problems in safety case maintenance, highlighting the need for a process for managing changes in safety cases. Firstly, one must understand the importance of challenges made to a safety case. Such challenges can question the validity of the safety argument and evidences that justify the argument. Secondly there is a requirement for identifying and understanding the impact that changes will make to the safety case structure. As the elements of a safety case are inter-dependant, minor changes can have major impacts to

---

\* Corresponding author: School of Informatics, The University of Edinburgh, Informatics Forum, 10 Crichton Street, Edinburgh EH8 9AB, UK, tel. +44-131-6505899, email: mfelici@inf.ed.ac.uk.

<sup>1</sup> Adelard Web Site: <http://www.adelard.com>

the overall structure of a safety argument. Thirdly, those in charge of maintaining the safety case must decide on the best course of action for strengthening the safety case, providing explicit justification as to why the changes are being made, allowing reviewers to have implicit confidence and faith in the safety argument. Finally, there is a requirement for recording detailed information pertaining to safety cases, which help identify quicker the impact of changes on the overall safety argument.

It is thus evident that we require a process or tool for change management in safety cases, allowing for system developers, and safety analysts to correctly record, monitor and justify the changes to a safety case over time, hence strengthening the validity of the safety argument.

Considering the GSN, which provides a graphical means of documenting safety cases, change management at its simplest involves three main structural changes that can occur in the safety case lifecycle:

1. Adding a node (e.g., adding a new claim or evidence)
2. Deleting a node (e.g., deleting a claim or evidence)
3. Modifying a node (e.g., modifying a claim or evidence).

One can capture these changes using propositional modal logic statements [1][2]. This formal representation can allow one to perform analysis on the safety case, allowing reviewers to begin to justify the changes during the evolution of the safety argument.

### **3. SUPPORTING SAFETY CASE MAINTENANCE**

This project concentrates on the ASCE tool v3.0, developed by Adelar, which provides a graphical means (using GSN) for development, review, analysis and dissemination of safety cases. The goal is to develop a plugin which can facilitate the change management process, allowing for safety reviewers to analyse the safety case, and visually monitor the impact of changes on the safety argument over time. The plugin will allow an ASCE tool user to maintain and view the history of structural changes that have been made to a safety case argument over time.

At present, the ASCE Difference Tool <sup>2</sup> allows the comparison of two safety case trees, documented in GSN. The tool only allows for comparison of two structured safety cases, without supporting the comparison of subsequent safety cases. It provides limited support for determining and analysing the evolution of a safety case argument over time. This highlights the requirement for having a tool which can maintain a history of changes made to the safety case argument, and provide a means of highlighting these changes to the user, hence more importantly allowing for safety reviewers to analyse and justify the impact of changes over time to the safety argument, thus strengthening the overall safety justification and argument.

### **4. DISCUSSION AND CONCLUSIONS**

To summarise, through development of this plugin for ASCE the objective is to allow safety reviewers to better understand the evolution of a safety case argument through the change process, and provide a means of managing the changes and their impact to the safety case. Through providing a means of capturing the structural changes of a safety case during each phase of the change process, this will improve the understanding of the resultant safety case, allowing analysts to better justify the validity of the final safety case. This will also support the formal representation of safety case changes in propositional modal logic form [2], hence allowing safety case developers to provide a formal justification for the changes made. Overall this will strengthen the final safety argument, through allowing safety analysts to determine and analyse the exact sequence of safety case changes that led to the construction of the final safety case.

It is intended that this plug-in be integrated into the ASCE tool, providing users with a means of maintaining a history of safety case changes, and seeing the structural impact of safety case changes over time. The plug-in will thus facilitate another dimension to safety case analysis, providing users with a better means of justifying and validating the strength of a safety argument. The tool will allow analysts to better understand the process of safety case evolution, thus enhancing safety case evolution practise amongst safety case users.

### **REFERENCES**

- [1] Felici, M (2006) Capturing emerging complex interactions: Safety analysis in air traffic management. *Reliability Engineering & System Safety*, 91, 1482-1493.
- [2] Felici, M (2006) Modeling Safety Case Evolution – Examples from the Air Traffic Management Domain. *Proceedings of RISE 2005, LNCS 3943*, pp. 81-96, Springer-Verlag.
- [3] Kelly, T.P and McDermid, J.A (2001) A systematic approach to safety case maintenance, *Reliability Engineering & System Safety*, 71, 271-284.
- [4] Kelly, T.P. and McDermid, J.A. (1999) A Systematic Approach to Safety Case Maintenance. *Proceedings of SAFECOMP'99, LNCS 1698*, pp. 13-26, Springer-Verlag.
- [5] Bishop, P. and Bloomfield, R. (1998) A Methodology for Safety Case Development. *Proceedings of Safety-critical Systems Symposium*.

---

<sup>2</sup> [http://www.adelard.com/web/hnav/ASCE/tools/asce\\_difference\\_tool.html](http://www.adelard.com/web/hnav/ASCE/tools/asce_difference_tool.html)