**Interner Bericht**

# DLR-IB-FT-BS-2025-058

# Large Language Models (LLMs) to Support Systematic Safety Assessments

## Hochschulschrift


Dietrich Kevin Zambou Zambou

Deutsches Zentrum für Luft- und Raumfahrt

Institut für Flugsystemtechnik
Braunschweig


Deutsches Zentrum für Luft- und Raumfahrt

Institutsbericht
**DLR-IB-FT-BS-2025-058**

# Large Language Models (LLMs) to Support Systematic Safety Assessments

Dietrich Kevin Zambou Zambou

Institut für Flugsystemtechnik
Braunschweig

Deutsches Zentrum für Luft- und Raumfahrt e.V.
Institut für Flugsystemtechnik
Abteilung Sichere Systeme & Systems Engineering

**Stufe der Zugänglichkeit: I, Allgemein zugänglich: Der Interne Bericht wird elektronisch ohne Einschränkungen in ELIB abgelegt.**

Braunschweig, den 11.04.2025

Unterschriften:

[redacted]

Institutsleitung: Prof. Dr.-Ing. S. Levedag

Abteilungsleitung: Andreas Bierig

Betreuer:in: Alexander Ahlbrecht

Verfasser:in: Dietrich Kevin Zambou Zambou

TU Clausthal

# LARGE LANGUAGE MODELS (LLMS) TO SUPPORT SYSTEMATIC SAFETY ASSESSMENTS

MASTER'S THESIS

presented by

DIETRICH KEVIN ZAMBOU ZAMBOU

Clausthal University of Technology
Institute for Informatics
Big Data and Technical Information Systems

Student ID [redacted]

Dietrich Kevin Zambou Zambou: *Large Language Models (LLMs) to Support Systematic Safety Assessments*

STUDENT ID

[redacted]

ASSESSORS

First assessor: apl. Prof. Dr.-Ing. habil. Umut Durak
Second assessor: Prof. Dr. Sven Hartmann
Supervisor: M. Sc. Alexander Ahlbrecht

DATE OF SUBMISSION

February 3, 2025

## EIDESSTATTLICHE VERSICHERUNG

Ich erkläre hiermit, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Alle Stellen der Arbeit, die wörtlich oder sinngemäß aus anderen Quellen übernommen wurden, wurden als solche kenntlich gemacht. Die Arbeit wurde in gleicher oder ähnlicher Form noch keiner anderen Prüfungsstelle vorgelegt.

*Clausthal-Zellerfeld, den 3. Februar 2025* [redacted]

## PUBLICATION AGREEMENT

Ich erkläre mich mit der öffentlichen Bereitstellung meiner Master's Thesis in der Instituts- und/oder Universitätsbibliothek einverstanden.

*Clausthal-Zellerfeld, den 3. Februar 2025* [redacted]

# ABSTRACT

In this thesis we will explore the application of Large Language Models (LLMs) to automate the generation of Safety Cases (SCs) in the Goal Structuring Notation (GSN) format. This thesis proposes a new framework that integrates a pretrained LLM (Llama-3-8B), fine-tuned with domain-specific prompts and datasets, to generate automatically SCs written textually in XML and using GSN. These SCs are generated based on supplied safety analyses and system descriptions. These SCs can be visualized as GSN diagrams within an application, facilitating better comprehension and verification by safety engineers.

A core innovation of this master's thesis is the creation of structured XML syntax to represent GSN elements and their relationships. These XML syntaxes were used to create full examples of SCs, which were then used to generate prompts to create the dataset that forms the basis for training the LLM. The model has been evaluated using the U-SHIFT autonomous vehicle from DLR (German Aerospace Center), a real-world example of safety-critical systems. The overall system was manually decomposed into manageable sub-systems. This enabled the LLM to generate SCs for each sub-system of the U-SHIFT Vehicle (onboarding, communication and control sub-systems). Results showed that the model was able to produce modular and SCs, corresponding well to provided safety analysis and system descriptions in the tested scenarios. It has to be noted, however, that testing was limited in scope, and it might be that performance of the model in bigger and more complex projects will bring to light further challenges, including possible inaccuracies and inconsistencies in the generated outputs.

The results highlight the potential of LLMs to streamline the development of SCs, provide a consistent starting point for project initiation, reduce manual effort and improve consistency in safety engineering practices. However, the study also identifies challenges, including token limitations for large-scale systems and the need for explicit regulatory compliance validation against regulatory standards such as ISO 26262, SOTIF, etc. Future research directions include extending the framework to support hierarchical SCs for highly complex systems, enhancing the GSN schema (creating an XML syntax for extension modules), and integrating real-time compliance verification mechanisms. This work try to highlight the potential of AI-driven methods in safety engineering by introducing an approach that could reduce the problems related to scaling up and efficiency in order to guarantee the reliability of safety-critical systems.

# ZUSAMMENFASSUNG

Diese Masterarbeit untersucht die Verwendung von Large Language Models (LLMs) zur automatisierten Erstellung von Safety Cases (SCs) im Goal Structuring Notation (GSN). Diese Masterarbeit schlägt ein neues Framework vor, das ein vortrainiertes LLM (Llama-3-8B) integriert, welches durch domänenspezifische Prompts and Datensätze feinabgestimmt wurde, um SCs textuell im XML-Format unter Verwendung von GSN zu erzeugen. Das trainierte Modell bekommt als Eingabe die Sicherheitsanalyse plus die Systembeschreibungen und generiert als Ausgabe GSN-basierte SCs im XML-Format. Danach können diese SCs direkt als GSN-Diagramme innerhalb einer Anwendung visualisiert werden, was ihre Verständigkeit und ihre Überprüfung durch Safety Engineers erleichtert.

Eine Kerninnovation dieser Arbeit ist die Entwicklung einer strukturierten XML-Syntax zur Darstellung der GSN-Elemente und ihrer verschiedenen Beziehungen. Diese XML-Syntaxen wurden dann verwendet, um vollständige Beispiele von SCs in XML-Format zu erstellen, die dann zur Erstellung von Prompts verwendet wurden, um den Datensatz zu erstellen, der die Grundlage für das Training des LLMs bildet. Das Modell wurde anhand des autonomen Fahrzeugs U-SHIFT des Deutschen Zentrums für Luft- und Raumfahrt (DLR) evaluiert, einem realen Beispiel für sicherheitskritische Systeme. Das Gesamtsystem wurde manuell und in handhabbare Subsysteme zerlegt. Dadurch war das LLM in der Lage, SCs für jedes der Subsysteme des USHIFT-Fahrzeugs (das Onboarding-, Kommunikations- und Kontrollsystem) zu generieren. Basierend auf den verschiedenen Tests und Analysen kann also angenommen werden, dass das Modell in der Lage ist, modulare und sicherheitsrelevante SCs zu erstellen, die gut zu den Sicherheitsanalysen und Systembeschreibungen passen, die in den getesteten Szenarien geliefert wurden. Es ist jedoch darauf hinzuweisen, dass die Tests in einem sehr eingeschränkten Anwendungsbereich durchgeführt wurden. Bei umfangreicheren, komplexeren und größeren Projekten weist das Modell wahrscheinlich weitere Probleme auf, wobei die erzeugten Ergebnisse wahrscheinlich ungenauer und inkonsistenter sind.

Die Ergebnisse betonen das Potenzial der LLMs, SCs zu optimieren, eine konsistente Ausgangsbasis für Projektinitiativen zu bieten, den manuellen Aufwand zu reduzieren und die Konsistenz in sichercheitstechnischen Verfahren zu verbessern. Gleichzeitig identifiziert die Studie Herausforderungen, darunter Token-Beschränkungen bei groß angelegten Systemen und die Notwendigkeit einer expliziten Validierung der regulatorischen Konformität gemäß Standards wie ISO 26262, SOTIF und andere. Zukünftige Forschungsschwerpunkte umfassen die Erweiterung des Frameworks zur Unterstützung hierarchischer SCs für hochkomplexe Systeme, die Verbesserung des GSN-Schemas (einschließlich der Entwicklung einer XML-Syntax fur Erweiterungsmodule) und die Integration von Mechanismen zur Echtzeitüberprüfung der Konformität. Diese Arbeit versucht, das potential KI-gestützter Methoden im Bereich der Sicherheitsingenieurwissenschaften hervorzuheben, indem ein Ansatz vorgestellt wird, der die Probleme im Zusammenhang mit Skalierbarkeit und Effizienz verringern könnte, um die Zuverlässigkeit sicherheitskritischer Systeme zu gewährleisten.

# CONTENTS

## LIST OF FIGURES



## LIST OF TABLES

# LIST OF LISTINGS

# LIST OF ABBREVIATIONS

**GSN** Goal Structuring Notation
**SCs** Safety Cases
**LLMs** Large Language Models
**XML** eXtensible Markup Language
**JSON** JavaScript Object Notation
**YAML** Yet Another Markup Language
**AI** Artificial Intelligence
**NLP** Natural Language Processing
**RAG** Retrieval Augmented Generation
**LoRA** Low-Rank Adaptation
**QLoRA** Quantized Low-Rank Adaptation
**PEFT** Parameter-Efficient Fine-Tuning
**CUDA** Compute Unified Device Architecture
**API** Application Programming Interface
**DNN** Deep Neural Networks
**SACM** Structured Assurance Case Metamodel
**CAE** Claim-Argument-Evidence
**ACG** Assurance Case Generation
**GRL** Goal-oriented Requirement Language
**MDE** Model-Driven Engineering
**UI** User Interface
**NIM** NeMo Inference Microservices
**API** Application Programming Interface
**TRL** Transformer Reinforcement Learning
**CPS** Cyber-Physical Systems
**RE** Requirement Engineering
**BnB** BitsAndBytes

# 1 INTRODUCTION

In this chapter we will present the motivation for this research and outline the various research questions that will form the basis of this study. After that, we will highlight the importance of automating the generation of safety cases using Large Language Models (LLMs) to improve the efficiency and reliability of safety-critical systems. We will conclude with an overview of the thesis structure, guiding the reader through the organization of the work.

## 1.1 MOTIVATION

In industries where safety is very important, like aviation, aerospace, automotive, healthcare, railway, thorough safety assessments are needed to reduce risks and avoid serious failures. These assessments usually include the creation of Safety Cases (SCs). SCs are structured arguments supported by evidence, that aim to demonstrate the safety and reliability of a system within a specific operational context [37]. However, creating SCs requires a lot of resources, needs a lot of manual work, time, and skill [42], [56]. Nevertheless, with Artificial Intelligence (AI), especially LLMs, there are now new opportunities to support these processes.

Popular LLMs, like Llama-3.1-405B created by Meta, show strong skills in Natural Language Processing (NLP) and text creation. Those LLMs are used in many fields and task because of these capabilities. Software developers use LLMs to make code and fix errors, translators use them for good translations, and content makers use them for marketing writing and scripts. This thesis looks at how to put LLMs into the important area of system safety to help create safety cases, making the process faster while keeping accuracy and thoroughness high.

In this thesis, we leverage LLMs, specially our fine-tuned version of the Llama-3-8B model, to automatically generate GSN-based safety cases in XML format. This work uses LLMs to support Safety Engineers in their daily activities, with the idea of overcoming some of the most important challenges in safety engineering: reducing the manual workload, minimizing human error, and enabling faster iteration cycles. This master's thesis will contribute to the development of AI-assisted tools that can be used in the safety engineering process.

## 1.2 RESEARCH QUESTIONS

This thesis addresses the following research questions:

1. How can LLMs be applied to support the development of safety cases ?
2. How could methodologies to generate safety cases and their graphical representation in GSN using LLMs look like ?
3. How can the findings be applied to the automotive domain, particularly in the context of an automated vehicle (U-SHIFT Vehicle from DLR) ?

## 1.3 STRUCTURE OF THESIS

This master's thesis has six chapters, along with an appendix and bibliography, to facilitate a thorough review of the research, its methodology, and findings:

**Chapter 1, introduction:** In this chapter, we show the motivation of the study, articulate the research questions, and provide an overview of the thesis structure. This introduction will then set the context for the use of LLMs in SCs generation and highlights the significance of the methodology presented.

**Chapter 2, theoretical background:** This chapter will introduce the necessary concepts on which the research is based. We start by covering the definition, importance, and structure of SCs. Here we will focus on the Goal Structuring Notation (GSN) and its XML representation. The chapter further provides an detailed overview of Large Language Models (LLMs), covering their architecture, training methods, and applications in technical domains. Finally, we will be looking at challenges with LLMs and potential for future developments in engineering contexts.

**Chapter 3, related work:** This chapter will review previous work relevant for this master's thesis. We then take a look at previous solutions for automated SCs generation, applications of LLMs in safety-critical domains, and LLM for fine tuning techniques. The final part chapter is an overview of the opportunities and limitations that LLMs present when facing the challenges within safety engineering.

**Chapter 4, methodology:** This part of the master's thesis contains the methodologies on how to generate SCs in XML format using our fine-tuned Llama-3 model. It first gives an overview of the approach, followed by description of the XML-based GSN representation used for safety cases generation. Then we will cover the dataset preparation, prompt engineering, and the configuration of Llama-3.1-405B for generating the dataset (both safety analyses and SCs in XML). After that, we will also discuss the fine-tuning process using the Unsloth framework, the deployment of the trained model on Hugging Face Hub, and the development of desktop application for XML-to-GSN conversion and visualization.

**Chapter 5, testing and evaluation:** In this chapter, we test the developed framework (especially the fine-tuned Llama-3-8B model) to a real-world example (the U-SHIFT autonomous vehicle from DLR, as a case study). It describes strategy and evaluates the model's performance on three selected subsystems: the Onboarding System, Control System, and Communication System. This chapter presents a safety analysis for each subsystem, along with generated XML safety cases and a comprehensive evaluation of the results. Finally, the comprehensive testing phase will be analyzed and some recommendations for future improvements will be given.

**Chapter 6, conclusion:** This chapter summarizes the main results of the work and points out its contributions to the safety engineering. It also reflects on the limitations of the methodology used and outlines future research opportunities, including improvements on scalability, validation of compliance, and generation of hierarchical SCs generation.

# 2 THEORETICAL BACKGROUND

This chapter provides the theoretical foundation for the thesis. It details the fundamental concepts of the SCs and the Goal Structuring Notation (GSN), together with their role in safety-critical systems. This also introduces LLMs and explains their architecture, training approaches, and use in various technical domains. These concepts form the groundwork for the methodology presented in Chapter 4.

## 2.1 SAFETY CASES AND GSN

### 2.1.1 *Definition and Importance of Safety Cases*

**Safety Cases** (SCs) *"is a comprehensive and structured set of safety documentation which is aimed to ensure that the safety of a specific vessel or equipment can be demonstrated by reference to: safety arrangements and organization; safety analyses; compliance with the standards and best practice; acceptance tests; audits; inspections; feedback; and provision made for safe use including emergency arrangements"* [37]. They serve as a comprehensive rationale for why a system can be considered acceptably safe to operate. The SCs are very important, especially in high-risk industries such as aerospace, automotive, nuclear power, and healthcare, since the system failures may lead to very critical consequences, including loss of life, environmental damage, or major economic loss [37].

Various international standards, such as ISO 26262 for automotive , ARP 4754 and ARP 4761 for aviation, and IEC 61508 for general functional safety, define objectives that should be met to ensure safety but do not specify methods to achieve them. However, organizations are given the freedom to choose suitable approaches, safety cases have been in widespread use in many domains as a way to provide evidence of compliance with defined objectives.

In practice, SCs for systems can become significantly large during their development and as they continue to evolve. For example, the initial SCs for airport surface surveillance operations spans approximately 200 pages [19], [52]. The manual creation of such extensive SCs can be time-consuming, and error-prone [42], [56].

There are many advantages of SCs: They provide a general framework to demonstrate that a system complies with all relevant standards and regulatory requirements. Furthermore, by employing meticulous documentation of safety-related factors, SCs facilitate an inclusive and thorough risk assessment. In addition, SCs provide a comprehensive record of the safety arguments and evidence, thereby rendering the safety assessment process transparent. And finally, SCs are not necessarily a requirement but can aid in the approval and certification of safety-critical systems.

SCs can be represented in a variety of formats, each with its own relative strengths and domains of application. Most common forms reviewed here include structured prose, graphical notations, and formal methods. Structured prose presents safety arguments in a textual format; it is often organized hierarchically and outlines the goals, strategies, and evidence. Although this format is accessible, it can become unwieldy for complex systems. Many graphical notations using various types of diagrams or schema, such as the GSN [58], Structured Assurance Case Metamodel (SACM) [54] , and Claim-Argument-Evidence (CAE), have gained popularity due to their visual clarity and ability to represent complex relationships succinctly. In this thesis, we will focus on the GSN representation because it is the most widely used today [12].

### 2.1.2 *Introduction to the Goal Structuring Notation*

**GSN** is a graphical notation, which is widely adopted today to represent SCs. GSN have been developed by Kelly and Weaver [36], and has become a de facto standard in many safety-critical sectors for visualizing and communicating complex safety arguments. GSN provides a way to present the relationships between safety goals, strategies for achieving those goals, and the supporting evidence. GSN improves the comprehensibility and review-ability of SCs, particularly for large-scale and complex-systems, by proving a clear, logical representation of safety arguments. GSN consists of several interconnected elements, each element serving a specific purpose in the construction of a comprehensive safety argument. It is very important to understand these elements and their relationships for effectively developing and analyzing SCs using GSN.

The key components of GSN are as follows: goal, strategy, context, solution, assumption, justification. Each element can be represented graphically. Additionally, every instance of these elements, that is created within a GSN diagram is assigned a unique identifier (ID) to distinguish it from other instances. Those instances within the GSN diagram are interconnected by two types of relationship: "InContextOf" and "SupportedBy". Below, we provide an in-depth description of each of the elements and their graphical representation.

### 2.1.3 *Example of goal structure*

Below is an example of Safety Cases in GSN.

▷ **A goal,** "rendered as a rectangle, presents a claim forming part of the argument" [28].

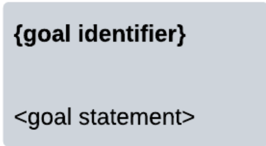


Figure 2.1: Graphical representation of a Goal in GSN

▷ **A strategy,** " rendered as a parallelogram, describes the inference that exists between a goal and its supporting goal(s)" [28].

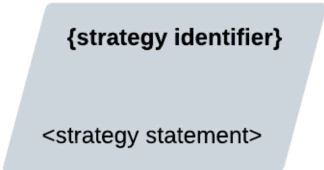


Figure 2.2: Graphical representation of a Strategy in GSN

▷ **A Solution,** "rendered as a circle, presents a reference to an evidence item" [28].

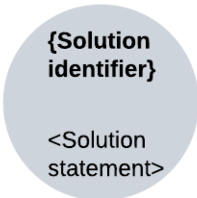


Figure 2.3: Graphical representation of a Solution in GSN

▷ **A context,** "rendered as show below, presents a contextual artifact. This can be a reference to contextual information, or a statement" [28].

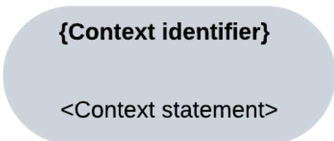


Figure 2.4: Graphical representation of a Context in GSN

▷ **A justification,** "rendered as an oval with the letter 'J' at the top- or bottom-right, presents a statement of rationale" [28].


Figure 2.5: Graphical representation of a Justification in GSN

▷ **An assumption,** "rendered as an oval with the letter 'A' at the top- or bottom-right, presents an intentionally unsubstantiated statement." [28].

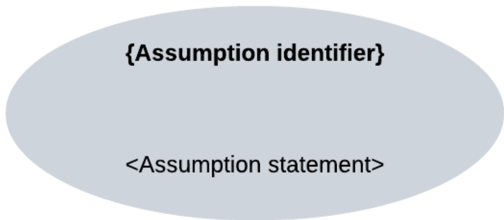


Figure 2.6: Graphical representation of an assumption in GSN

▷ **SupportedBy,** "rendered as a line with solid arrowhead, allows support relationships between elements to be documented. Permitted 'supported by' connections are: goal-to-goal, goal-to-strategy, goal-to-solution, strategy-to-goal" [28].

Figure 2.7: Graphical representation of a "SupportedBy" relationship

▷ **InContextOf,** "rendered as a line with a hollow arrowhead, declares a contextual relationship. Permitted 'in context of' connections are: goal-to-context, goal-to-assumption, goal-to-justification, strategy-to-context, strategy-to-assumption and strategy-to-justification" [28].

Figure 2.8: Graphical representation of a "InContextOf" relationship

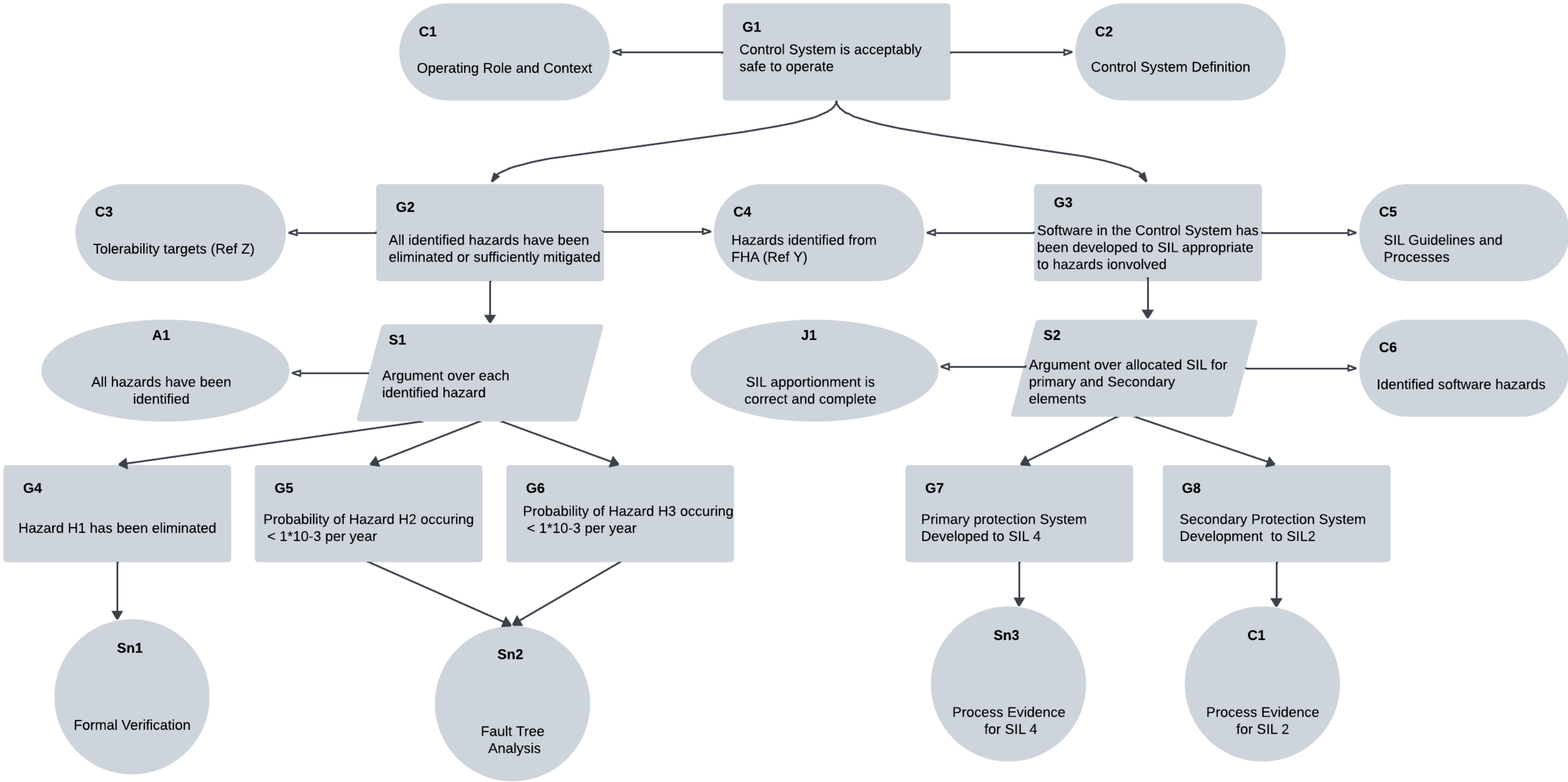


Figure 2.9: Example of safety cases in GSN [28]

## 2.2 INTRODUCTION TO LARGE LANGUAGE MODELS (LLMS)

### 2.2.1 *What are LLMs ?*

Large Language Models (LLMs) are a particular type of Artificial Intelligence (AI) Models for understanding and generating human language on a large scale are reviewed here. Usually built using deep learning techniques, such models are trained conventionally on large amounts of text data to understand the complexity and nuances of language [2], [71]. The underlying architecture of LLMs is based on transformers, and enables processing and generating text with a high degree of fluency and coherence [62].

One of the fundamental characteristics of LLMs is their ability to perform a wide range of (language-related) tasks without needing a specific task-specific training. This capability of LLMs to perform a wide-range of language related tasks comes from their training, that was accomplished on diverse datasets. For example, GPT-3, created by OpenAI, is capable of producing text at a level similar to humans (translating languages, summarizing long documents, holding a conversational dialog, etc.) [7]. To achieve such a high level of performance, training LLMs requires using huge amount of data [2]. During this training phase, the model predict the next words in a sentence to learn patterns and structures in language. The effectiveness of the transformer architecture, developed by Vaswani, Shazeer, Parmar, et al. [62], has been highly proved for various language-related tasks. At the core of this model is the self-attention mechanism that enables the model to have a unique ability in estimating the relative importance of words in a sentence. This allows the transformer to understand context more deeply so that it can generate much more coherent and meaningful text. In a nutshell, by analyzing these interrelations among words, the model learns to appreciate such subtleties in language that result in outputs that would be increasingly precise and contextually relevant across different language-related tasks. The training process consists in predicting the next word given a certain sentence.

### 2.2.2 *The History of LLMs*

Starting from the rule-based systems in the 1950s and 1960s, it is now a few-decade-long development process for LLMs. Though these old models were able to do magic in those days, they suffered from the disadvantage of their own hand-coded linguistic rules and features that limited the solution space of NLP [32], [39].

The 1980s and 1990s saw the introduction of Statistical Language Models using probabilistic methods for judging the likelihood of a word sequence in a given context. These models had achieved better accuracy and huge data handling. Compared to their rule-based predecessors, they had better capabilities but still fell short in understanding the fine-grained semantics and contextual aspects of language[15], [35].

A good path was developed starting in the mid-2010s with the implementation of Neural Language Models [4]. It utilizes the profound concept of deep learning in learning all structures of the extensive text corpus. The origins for this beginning point is the RNNLM introduced back in 2010; giving a much finer representation context that gave out more natural, coherent-like textual output [38].

Google launched in 2015 the Google Neural Machine Translation (GNMT) system, a pioneering large-scale neural language model. Trained on extensive bilingual data, GNMT achieved remarkable performance in machine translation tasks [69].

The transformer architecture, introduced in 2017 [62], radically improved this sub-field by allowing to model long-term language dependencies and concurrently train the models on many GPUs. This finding made way for the training of a much larger model Wolf, Debut, Sanh, et al. [67].

OpenAI's GPT-1, released in 2018 [50], represented a major milestone in natural language processing. With its transformer-based architecture and 117 million parameters, GPT-1 demonstrated the ability to generate contextually relevant sentences, showcasing the potential of transformers in NLP tasks [1].

GPT-3, released in 2020 by OpenAI, was considered as the largest LLM at that time. This model's ability to generate highly coherent and natural-sounding text across a wide range of NLP tasks highlighted the immense potential of LLMs [43], [25]. After the gigantic success of GPT-3 worldwide, OpenAI released in 2024 GPT-4.

After the gigantic success of GPT-3 worldwide, OpenAI released GPT- 4 in 2024. Meanwhile, though, open-source LLMs had taken up much traction. One would note, for example, the LLaMA family and its pivotal role in this movement. In February 2023, Meta AI released LLaMA 1, which was an instant hit with researchers and developers even in its initially somewhat constrained form. The development continue with the release of LLaMA 2 in July 2023. This new model version model is much more permissively licensed, allowing much broader commercial use.. This was followed by LLaMA 3, launched in February 2024, with improvements over previous versions. LLaMA 3 comes in three sizes: 8B, 70B, and the 405B model [24]. LLaMA 3.1 405 Billion parameter model, released on 23 July 2024, is considered one of the world's largest and most openly available foundation models. The 405B model demonstrates state-of-art capabilities in general knowledge, steerability, math, tool use, and multilingual translation.

### 2.2.3 *Architecture of a LLM (Large Language Model)*

LLMs are architecturally designed mostly on transformer architecture. Transformers have been among the breakthroughs in natural language processing since their release in 2017 by Vaswani, Shazeer, Parmar, et al. [62], [41].

One of the important components of this architecture has been the incorporation of an attention mechanism, where a model would be able to estimate how important different parts of input data are simultaneously. It permits rapid and

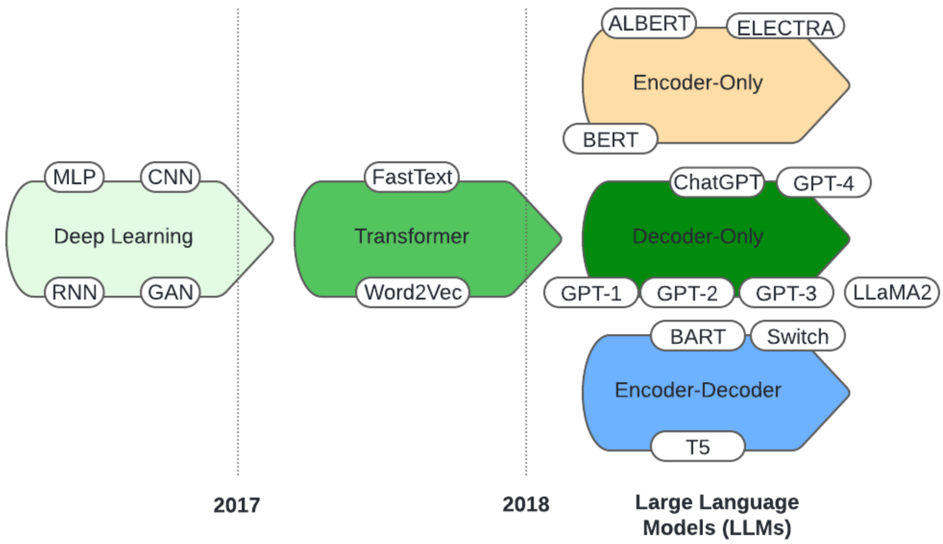


Figure 2.10: Evolution Roadmap of Large Language Models [32]

efficient processing of linguistic information that forms the backbone for pre-training LLMs. In pre-training, such models learn a great number of linguistic patterns from huge text datasets to build up a large knowledge base about linguistics transformer. There are two major components making up a transformer: the encoder and the decoder.

The transformer is composed of 2 sub-components: the encoder and the decoder. The encoder transforms input sequences into a set of high-dimensional continuous representations, while the decoder incorporates a self-attention mechanism; this enables the model to consider the whole sequence rather than each component separately. To address the model's non-recurrent nature, positional encoding are added to the input to preserve sequential order information [62], [53].

There are 2 main parts the transformer has: the encoder and the decoder. The encoder maps input sequences into a set of high-dimensional continuous representations, and the decoder has a self-attention mechanism, so the model can have consideration for the whole sequence rather than each element as what it is. To address the model's non-recurrent nature, positional encoding are added to the input to preserve sequential order information [62], [53].

The complexity and substantial number of parameters in LLMs greatly enhance their capacity to learn and capture diverse linguistic features and dependencies from large datasets. Models like those in the Llama 3 family, discussed by Dubey, Jauhri, Pandey, et al., have parameter that ranges from 1B to 405B. These parameters allow the model to have several layers of abstraction, hence being capable to perform complex tasks (translation, summarization, coding, and even conversation) with a high level of accuracy.But including more parameters causes higher computational burden and memory utilization, making these models

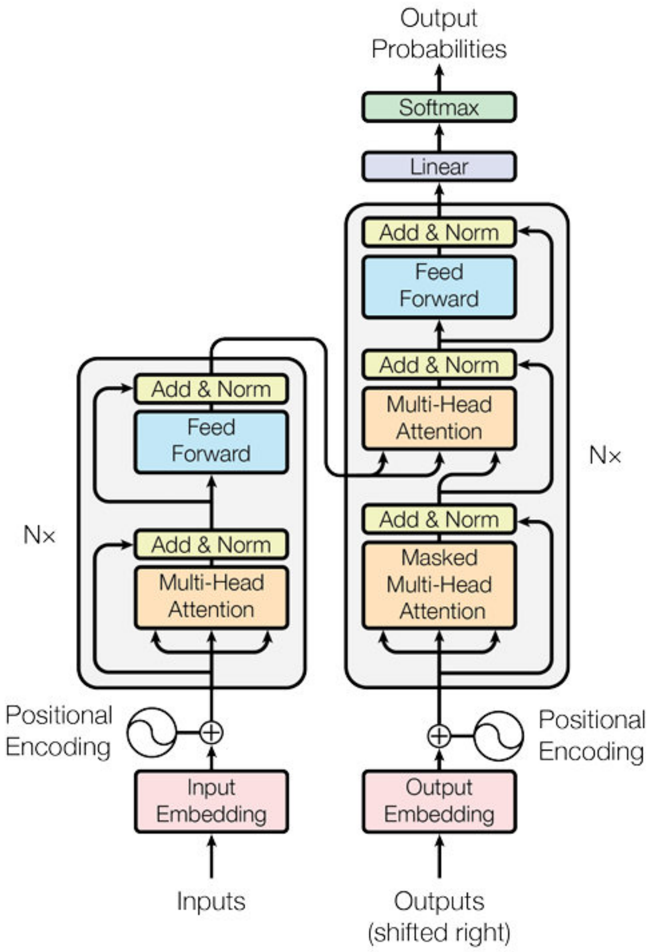


Figure 2.11: Model architecture of the Transformer (Vaswani, Shazeer, Parmar, et al.[62])

computationally heavy. So there should be a trade-off between the advantage of using extra parameters and the computation challenges of implementing these memory-consuming models in real-world applications. Consequently, several studies, for example, by Chavan, Magazine, Kushwaha, et al. [11], focus on compressing these models. Model compression refers to the reduction in size and computational requirements of Deep Neural Networks (DNN) without sacrificing much of their performance—an aspect that is quite important in the deployment of LLMs on low-resources devices such as mobile phones and embedded systems.

### 2.2.4 *The training steps of a LLM*

The Training of LLMs involves some necessary steps. Those steps enable effective learning from large textual data. Here, the major stages of LLM training are summarized (data preparation, model configuration, pre-training, fine-tuning, and evaluation) [22].

▷ **Identification of the target:** An LLM training process starts from the use case of the model, as the target determines the data sources for training the model. The target and the LLM use case continuously evolve. During this evolution, there is an adaption of new elements during the training and fine-tuning.

▷ **Data preparation:** After the identification of the target, we will collect and pre-process large text corpora related to this target. This step includes, but is not limited to, tokenization, data cleaning, and normalization - thereby preparing the data for training. High-quality data is very important because the model's performance is directly related to it.

▷ **Model configuration:** After preparation, the next thing is to define the model structure. Most LLMs base their architecture on transformer. This is the reason why, they need a definition of parameters (the number of layers, attention heads, and hyper-parameters). In addition, researchers often conduct experimental research using these parameters to optimize performance.

▷ **Pre-training:** The model, once configured, has to be exposed to a large corpus of data so that it can learn patterns of the language and context in which to apply them. Common objectives include masked language modeling and next-token prediction. This step requires a lot of computational power and typically requires significant resources because.

▷ **Fine-tuning:** This comes after the pre-training step. The model is fine-tuned on a smaller, which is task-specific. During this phase, the model is trained with supervised-learning for certain tasks such as text classification or question answering. This makes the model more useful for a very wide range of applications and improve its performance.

- ▷ **Evaluation:** This is the last step in modeling performance evaluation using appropriate metrics for tasks that it has been fine-tuned to perform. Commonly used metrics include accuracy, F1 score and BLEU score. The model evaluation should be performed thoroughly to ensure generalization to data that it has never seen before.

## 2.3 APPLICATION OF LLMS IN TECHNICAL DOMAINS

### 2.3.1 *Existing applications and research*

LLMs have applications in many technical fields: finance, mechanical engineering, mathematics, etc. [29].

- ▷ **Finance**

  LLMs are revolutionizing the finance industry, allowing a wide array of applications in the area of operational efficiency improvement and decision-making process enhancement. Advanced AI systems are being applied to several important financial domains.

  One notable example is BloombergGPT [68], a sophisticated LLM with 50 billion parameters. It has been trained on a huge and various financial data. This model has resulted in significant improvement in the financial NLP tasks. BloombergGPT can deliver an enormous boost to customer service in efficiently managing queries and creating top-shelf financial advisories.

  The other very vital area where LLMs are also making great strides is in risk assessment and management. In predicting risks and suggesting the way of mitigation, there are many financial algorithms the model use (historical data and market trends). Financial institutions are increasingly relying on LLMs to make credit risks assessments, loan approval decisions, and investment strategies in general.

  However, financial data has very sensitive features that demand more attention to privacy and security issues; In order to handle these challenges, methodologies like data encryption, and full data protection strategies have to be in place so that LMMs operate effectively while meeting the regulatory requirements.

  The development of FinGPT [70], a Large Language Model specifically designed for the finance sector, is a promising line of development in this regard. This will be continued, and even more specialized financial LLMs are likely to emerge as the research proceeds.

▷ **Mathematics**

The application of ChatGPT in mathematics education has shown great promise. Wardat, Tashtoush, AlAli, and Jarrah [65] found that ChatGPT could improve the teaching of mathematics. The latter is substantiated by the fact that AI models can create interactive and tailor-made learning experiences that will be able to produce custom examples and problem-solving methods based on the needs of every students. It allows for instant feedback, identification and critical areas where students are making mistakes, and possible means of approaching those areas. This means that ChatGPT can respond almost instantly, giving accurate identification of specific points where the learners are making mistakes and several targeted strategies in getting past these barriers. It is also capable of doing simple calculations and solving simplistic equations. Yet, it will be very unpredictable in its performances, given various factors such as the complexity of a problem, precise input information, or how well instructions are designed. Researches (Frieder, Pinchetti, Griffiths, et al. [26]) conducted an evaluation of its capabilities in mathematics to date with ChatGPT by testing it against a publicly available dataset. They compared its performance with other models specially developed for mathematical data (e.g. Minerva).

### 2.3.2 *Challenges of LLMs*

Although LLMs are very powerful tools with impressive capabilities of understanding and generating human-like texts, they have complex challenges and limitations, that we will present in this section: [29]

▷ **The massive computer resources required for their training and deployment**. Those LLMs count often billions of parameters. That is the reason why they necessitate massive computing infrastructure and energy consumption. Following researchers (Strubell, Ganesh, and McCallum) [59] noted this issue in their research work, estimating that training a single layer transformer model can emit as much carbon as five cars over their lifetimes. This issue naturally raises important concerns about the viability of developing large language models (LLMs) and widespread use, with researchers developing increasingly larger models in attempt to maximize performance.

▷ The following challenge involves their **high demand for data**. LLMs require a massive corpus, often drawn from the web, and therefore present complications in terms of data privacy, copyright infringement, and ethical concerns about utilizing sensitive and private information in a non-consensual manner.

Following researchers (Bender, Gebru, McMillan-Major, and Shmitchell [6]) have discussed at length such concerns, stating that both model performance and fairness depend a lot on diversity and quality in training data. If training data contain biases, this may lead to LLMs that, as been demonstrated, can produce discriminatory outputs.

- ▷ Another challenge is **the opacity in the decision-making process of LLMs**. This is commonly known as "black-box" problem. It offers very serious obstacles to the interpretability and accountability called for. Following researchers have investigated the interest in interpretability of LLMs by maintaining that the lack of transparency in LLMs is a barrier toward Understanding, debugging and trusting their outputs. The lack of interpretability then becomes a serious problem in the critical applications where it's very important that the reasoning behind a model's decision be explained and justified.

- ▷ Another limitation of Large Langue Models (LLMs) pertains to their **lack of long-term factual accuracy and coherence in long conversations or texts**. The study conducted by (Gallegos, Rossi, Barrow, et al. [27]) examined this problem and discovered that, even when LLMs have preformed well in short-term language modeling, they often lack accuracy and logical cohesion in long-term settings.

In the image below, we present the various challenges associated with LLMs in greater detail (figure 2.12).

This chapter has established the essential theoretical background, including GSN and LLMs, that underpin the methodology developed in this research. The insights gained here are critical for understanding the proposed approach. Chapter 3 will now review related work, identifying gaps in existing literature and justifying the need for this study.

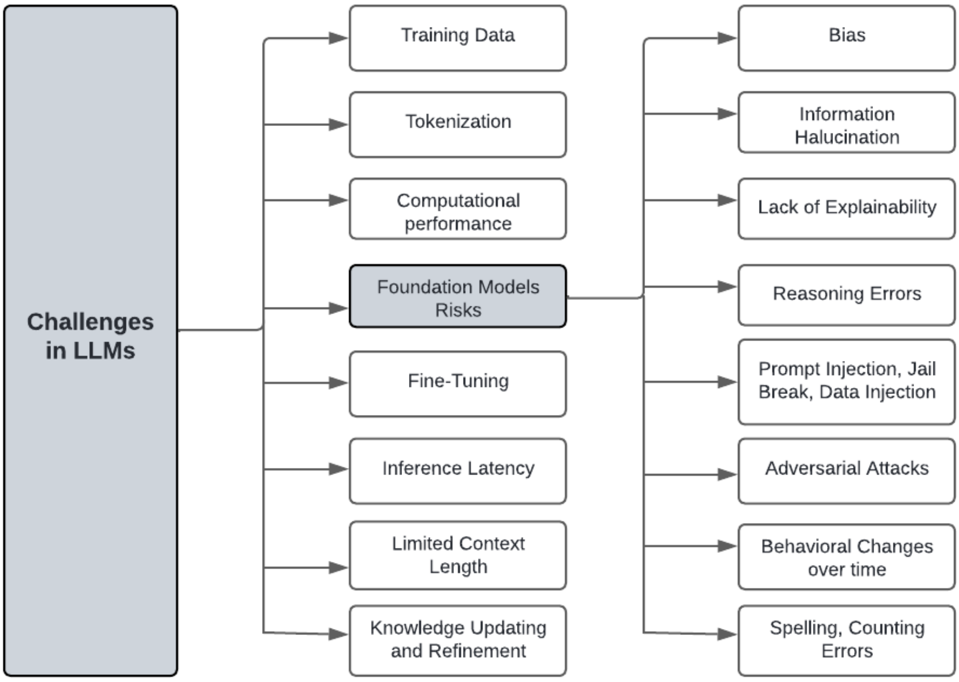


Figure 2.12: Challenges of LLMs (Hadi, Al Tashi, Shah, et al. [29])

# 3 RELATED WORK

This chapter examines the present landscape of research regarding automated SCs generation and the application of LLMs in safety-critical contexts. This analysis emphasizes current methodologies, their constraints, and the unresolved challenges that persist. This chapter identifies research gaps, thereby establishing the unique contribution of the thesis and its relevance.

## 3.1 AUTOMATED GENERATION OF SAFETY CASES

Interest in automation of the SCs generation process has recently been growing among researchers in the last years [55]. Several significant contributions have arisen, each presenting distinct methods to optimize and improve the creation and assessment of SCs. A notable advancement was made by researchers (Denney and Pai [18], in 2014) who developed a methodology for automatically generating SCs through formal verification techniques. This method was implemented in NASA's Swift Unmanned Aircraft System (UAS). The study lacked a mechanism for evaluating the reliability of the generated SCs.

A major development in this field is the introduction of AdVoCATE Denney and Pai [19], in 2018, developed and automated tool for improving the construction and evaluation of SCs. This innovative tool supports manual operations and embeds a variety of automated features. These include metadata integration, format translation, compatibility with other SCs tools, composition through auto-generated components, and the calculation of safety cases metrics. AdvoCATE primarily employs the GSN notation and can seamlessly merge manually created safety case elements with content from external tools, although this integration is currently limited to the AUTOCERT formal verification tool Denney and Pai [17]. AdvoCATE offers a comprehensive suite of functionalities, including: creation and assembly of safety assurance arguments, integration of formal methods into broader assurance arguments, instantiation of assurance patterns, hierarchical and modular abstraction, generation of views from user-defined queries, verification of argument structure properties [19].

In the erea of Cyber-Physical Systems (CPSs) researchers (Hartsell, Mahadevan, Dubey, and Karsai [30]) present an automated approach to create **AC!** (ACs). This technique uses patterns to minimize manual development of ACs that are error-prone, especially when dealing with connected system models. The study shows how this method enables the development of integrated ACs with source artifacts. It is important to acknowledge that the produced assurance cases may be deficient, as some nodes may be devoid of parameters and cannot be instantiated automatically.

Another significant contribution to the field is an automated framework for generating ACs (Wang, Oh, Low, et al. [64], which aids in the creation, validation, and assessment of SCs generation, it's important to note that their evaluation focuses solely on specific sections of the ACs that are either fully or nearly complete, which may not provide a fully accurate assessment of the overall SCs.

Ramakrishna, Hartsell, Dubey, et al. [51] conducted a research, that identified that traditional manual approaches to safety case development suffer from a lack of robustness and systematic methodology. To address this challenge, a novel tool called the Assurance Case Generation (ACG) was proposed. The AGC tool uses design artifacts, existing evidence, and developer experience to automate the construction of SCs. The AGC tool uses design artifacts, pre-existing evidence, and the experience of developers to automate to enable the automated generation of software components. The AGC is a great success in the development of software components. The authors acknowledge various limitations in their approach. For a start, many areas still require human intervention. Specially, human labor is still needed for translating certification standards into GSN. Moreover, today's SCs evaluation is based on one metric and may not be an all-inclusive assessment.

## 3.2 LEVERAGING LLMS FOR THE GENERATION OF SAFETY CASES

Researchers (Weyssow, Sahraoui, and Syriani [66]) conducted a research to address the challenges in meta-models, which is a way to underline correlations among notions in the field of Model-Driven Engineering (MDE). A method based on Deep-Learning using pretrained LLMs has been proposed. The researchers showed that it is able to effectively provide recommendations for scenarios of renaming concepts.

The advanced capabilities of LLMs such as GPT-4 in the Requirement Engineering (RE) concept has been showcased in another recent work by Chen, Chen, Hassani, et al. [14] particularly focusing in the development of goal-oriented models. The findings show that GPT-4's performs well in to the Goal-oriented Requirement Language (GRL). However, some generated elements were found to be generic or incorrect. Nonetheless, the concepts produced were beneficial, especially for stakeholders who might not be well-versed in the domain. This paper builds upon and expands this research.

Additionally, Chaaben, Burgueño, and Sahraoui [9] and Cámara, Troya, Burgueño, and Vallecillo [8] have utilized ChatGPT for generating Unified Modeling Language (UML) models. An innovative approach to enhance domain modeling is presented. This method utilizes LLMs without requiring training during a very-long time.

Their method utilizes few-shot prompt learning, thereby obviating the necessity for extensive datasets in training of the model. Simultaneously, the latter examines the implications of LLMs such as ChatGPT, highlighting their strengths and limitations in software modeling tasks, especially in UML modeling.

Viger, Murphy, Diemert, et al. [63] proposed the utilization of generative AI to identify potential objections, or defeaters, in ACs to improve their reliability. This work is preliminary and lacks empirical validation. However, it suggests a bright avenue for future research.

Similarly, Sivakumar, Belle, Shan, and Shahandashti [56] investigated GPT-4 for SCs generation, which is an essential activity in safety certification of safety-critical systems (autonomous vehicles and cyber-physical systems). The researchers concluded that GPT-4 performed with a level of accuracy in generating safety arguments, and with semantic coherence with regard to currently safety standards. In spite of GPT-4's capabilities, the authors stressed supervision, citing, for instance, that incorrect information and even fabricated information could occur in individual arguments.

## 3.3 LLMS FOR SAFETY ANALYSES

LLMs have been employed in transportation for safety analysis in highway construction and aviation. Researchers have investigated the use of LLMs to analyze highway construction safety -Smetana, Salles, Sukharev, and Khazanovich[57] - 2024, leveraging their NLP capabilities to extract insights from safety-related data. In the aviation sector, a case study examined the potential of generative language models for safety analysis using the Aviation Safety Reporting System (ASRS) (Qi, Zhao, Khastgir, and Huang[49]), demonstrating the models' ability to process and analyze large volumes of safety reports.

In the same movement, other potential applications of LLMs within safety analysis methodologies has also been investigated. A tangible example is a research on the system-Theoretic Process Analysis (STPA) conducted by Qi, Zhao, Khastgir, and Huang[49]. This study explored the use of ChatGPT in order to demonstrate the potential of LLMs in supporting safety analysts across various industries. The findings showcased the possibility of using LLMs to enhance the efficiency and effectiveness of safety assessment processes.

But, we notice that these studies highlight the flexibility and adaptability of LLMs in different safety domains, but they often rely on general-purpose prompt engineering.

## 3.4 LLM FINE-TUNING

The research on LLMs fine-tuning has evolved considerably [48]. The need for models that can be adapted to specific tasks in a large number on domain is also increasing. We learned in the last chapter, that LLMs are initially pre-trained over large datasets. After that, the models move on to the transfer learning phase, which can further be classified under two principal approaches: the feature-based transfer and the fine-tuning methodologies. Transfer learning is particularly useful when the available dataset is insufficient to train a model from scratch. Consequently, allowing using pretrained weights as a starting point. Under the category of feature-based transfer learning, features extracted from the pretrained model are adopted by a relatively small model dedicated to the targeted domain dataset. The models then move on to the transfer learning phase, which can further be classified under two principal approaches: the feature-based transfer and fine-tuning methodologies. Transfer learning is particularly useful when the available dataset is insufficient to train a model from scratch, thus allowing using pretrained weights as a starting point. Under the category of feature-based transfer learning, features extracted from the pretrained model are adopted by a relatively small model dedicated to the targeted domain dataset. Conversely, fine-tuning involves adjusting the pretrained weights on a task-specific dataset, with various techniques such as adapter tuning, gradual unfreezing, and prompt tuning, depending on how many layers are fine-tuned and the management of prompts during this process.

The Retrieval Augmented Generation (RAG) approach serves as an alternative to fine-tuning. It entails chunking documents, transforming them into embeddings, and storing these embeddings in a vector database for retrieval via similarity searches. This method is frequently regarded as a viable production solution because of its efficiency and reasonably accurate outcomes Jeong [34].The effectiveness of RAG may be considerably compromised by inadequate retrieval mechanisms. For instance, while some studies [16] suggest that adding noisy documents can enhance information retrieval in RAG, the quality of the responses remains constrained by the limitations of the similarity search, rather than the capabilities of the LLM itself.

Comparative studies [5] indicate that fine-tuning can yield more concise and accurate outputs than RAG pipelines, as evidenced by experiments conducted on domain-specific data, such as those from agricultural journals. However, the authors also note the considerable initial investment required for fine-tuning.

Recent advancements, particularly in Low-Rank Adaptation (LoRA) [31], have revolutionized fine-tuning by allowing the adjustment of a limited number of essential parameters-often in the range of thousands to millions-rather than the billions typically found in full model parameters. Additionally, the exploration of quantization techniques has opened new pathways for training LLMs on resource-contrained systems, enabling effective training at reduced memory costs[20]. The integration of quantized LLMs with parameter-efficient techniques like LoRA has demonstrated that satisfactory results can be achieved with limited resources [21].

Parameter-Efficient Fine-Tuning (PEFT) is a method used in Natural Language Processing (NLP) to improve the performance of pre-trained language models on targeted downstream tasks. PEFT conserves computation resources and reduces training time by reusing parameters from a pretrained model and fine-tuning them on smaller datasets, rather than starting from scratch. This technique typically involves freezing some layers of the pre-trained model while adjusting only the final layers relevant to the task. Various PEFT techniques include: Adapter, IA3, Prefix tuning, P-tuning, Prompt tuning, LoRA and QLoRA. But we will focus on the methodologies of the last two in this thesis

LoRA [31], a fine-tune technique LLMs by adding small, trainable sub-modules next to the feed-forward layers in the transformer architecture. These modules use rank decomposition to drastically decrease the number of trainable parameters while maintaining or enhancing performance. LoRA introduce two additional feed-forward layers: one that projects the input into a lower-dimensional space and another that restores it to the original size. This results in an updated representation that minimizes task-specific parameters, facilitating efficient task-switching and reducing hardware demands without increasing inference latency.

Quantized Low-Rank Adaptation (QLoRA) is an enhanced version of LoRA that reduces the precision of weight parameters to 4 bits. This reduction in precision decreases the model size, making QLoRA particularly useful in scenarios where memory resources for fine-tuning are limited [45].

## 3.5 CONCLUSION FROM RELATED WORK

Current researches indicates that LLMs have significant potential for generating SCs in various technical fields. Nonetheless, a prevalent limitation in these studies is their dependence on prompt engineering, wherein users strive to obtain precise responses from the model based in its general knowledge. We can notice that these methods are highly flexible. However, they lack the necessary precision and depth for safety-critical applications. Through this master's thesis, we try to propose a new framework to address these challenges by:

1. Developing a special XML syntax for GSN-based SCs to enable structured text generation
2. Converting textual SCs in XML into GSN diagrams,
3. Training a LLM on domain-specific datasets (including SCs, Safety Analysis and System design, which are closer to the real-word) so that the model outputs are directly tailored for the task without relying on prompt-engineering. This way, it could lead to substantial improvement in accuracy, reliability, and applicability of automated SCs generation, resulting in a large improvement over current approaches in this field.

# 4 METHODOLOGY

This chapter presents the methodology for generating SCs using a fine-tuned Llama-3-8B model. It describes the creation of a structured XML representation for GSN elements, dataset analyses and system descriptions. Additionally, the chapter details the development of a desktop application to visualize the SCs and GSN diagrams. The methodology builds on the theoretical concepts introduced in Chapter 2 and addresses the gaps identified in Chapter 3.

## 4.1 INTRODUCING THE METHOD

The developed methodology consists of 6 main components:

1. **The safety engineer:** Responsible for conducting the operations.

2. **Our trained LLM:** we fine-tuned the Llama-3-8b-bnb-4bit (from HuggingFace Hub) on our own data.

3. **The safety analysis and system description/design**.

4. **A desktop Application:** Developed in Python using the framework Tkinter, that application allows us to interact with the model and access the results.

5. **XML output:** generated by the LLM.

6. **GSN Diagram:** generated by the program.

Below, we present the the complete and end-to-end process, involving each of the above components and highlighting how they interact.

1. The Safety Engineer generates or first establishes the safety analysis and system description, which will then be saved in a file.

2. He then transmits the contents of this file to the LLM ( fine-tuned Llama-3-8b-bnb-4bit) via the main user interface of our program to the LLM and clicks on the "Generate GSN Diagram" button on the UI.

3. The application calls our LLM via an API to sends it the SCs and system description provided by the safety Engineer.

4. The LLM takes the SCs and the System description as input

5. The LLM will then be able to generate XML SCs in compliance with GSN, and corresponding to the variety of examples provided by the dataset during the training phase. When the SC in XML are generated, the program saves them and then immediately convert them into a GSN diagram using the Python package called Graphviz which facilitates the creation of diagrams in Python language.

6. Finally, the GSN diagram can be visualized in a separate window from the main window and, if desired, the SCs can be viewed in XML.

7. The Safety Engineer The safety Engineer, if he wishes, must click on "view in XML" to be able to visualize SCs in XML representation, displayed in a pop-up window. The Safety Engineer can also click on "Regenerate GSN-Diagram" to regenerate the SCs using the same input. He can also download the generated GSN-diagram in PDF, once he is satisfied with the results.

Below is the diagram of the developed method.

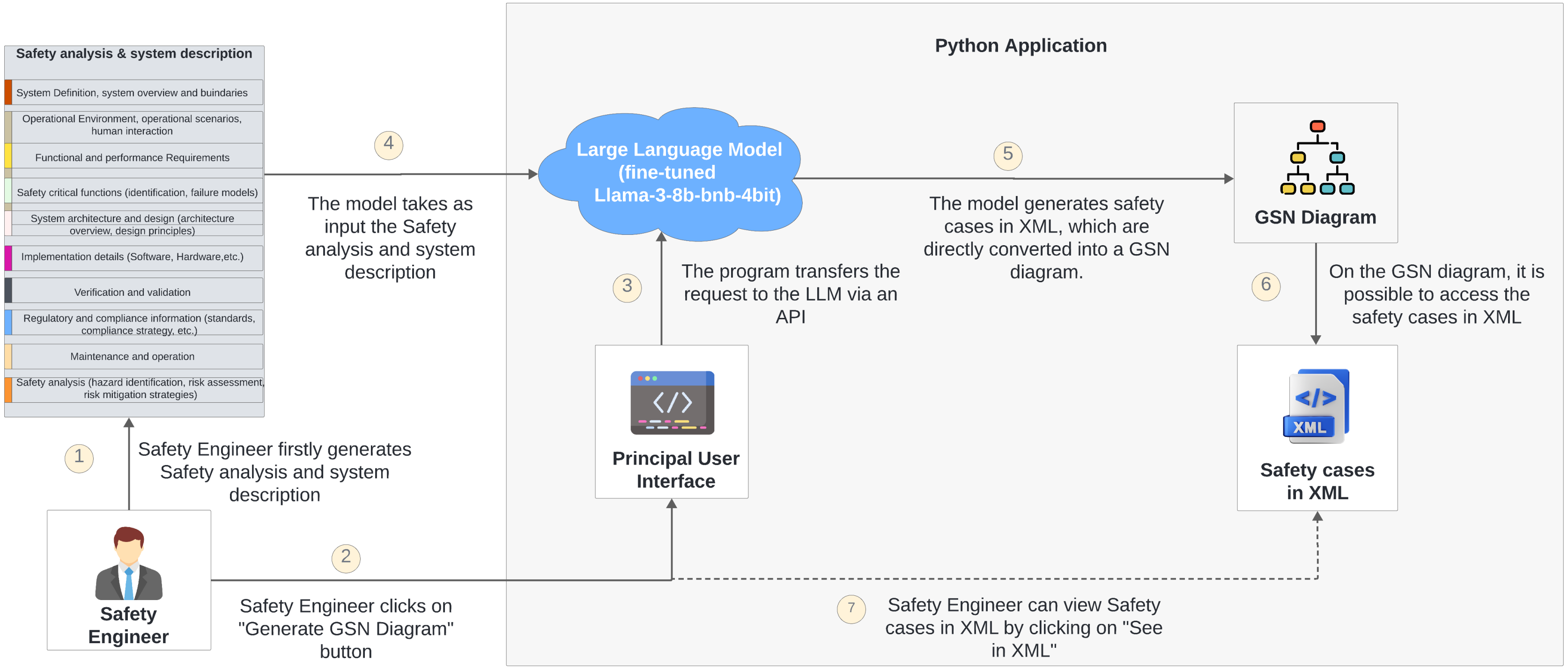


Figure 4.1: Overview of the developed method

In the rest of this chapter, we'll look in depth at how we represent our GSN-based SCs in XML, how we write our system prompts, how we prepared our dataset to train our LLMs, how we fine-tuned Llama-3-8b-bnb-4bit on our own generated dataset, how the APIs used to communicate with the LLMs look like, which fine-tuning techniques we have used, and how our application looks like.

## 4.2 REPRESENTATION OF SAFETY CASES IN XML ACCORDING TO GSN NOTATION

eXtensible Markup Language (XML) is a format that in the last years emerged, because it is a widely used format for representing structured data due to its flexibility and readability. When it comes to represent textually GSN-based SCs, XML provides an excellent textual format to organize, manage, and exchange safety-related data across different systems and stakeholders. This format has been adopted for the representation of data across many domains (safety-critical systems, document management, and web services, etc.). It provides a platform-independent way to define and share data in a hierarchical, human readable, and machine-processable format. This chapter introduces the method for representing GSN-based safety cases in XML, followed by a detailed breakdown of each GSN element and concludes with a comprehensive XML template of a SCs.

Other formats, such as JavaScript Object Notation (JSON) and Yet Another Markup Language (YAML), could be used to represent SCs in textual form. But the table below (Table 4.1) shows why we chose XML over others [3].

Table 4.1: Comparison of XML Advantages over JSON and YAML for Safety Case Representation

| **Advantage** | **XML** | **JSON** | **YAML** | |
|---|---|---|---|---|
| **Hierarchical structure** | Excellent | Good | Good | XML's tree-like structure manually aligns with complex, nested SCs elements, making it easy to represent hierarchical relationships such as goals, strategies and contexts. JSON and YAML can handle hierarchy but are less explicit and readable in deeply nested structures. |
| **Data validation (XSD)** | Robust | Limited | Minimal | XML supports robust data validation through XML schema (XSD). This ensures that the SCs conforms to predefined rules. JSON has limited validation with JSON schema. YAML offers very basic validation. |

| | | | | |
|---|---|---|---|---|
| **Attribute for Metadata** | Yes | No | No | XML allows the use of attributes (e.g.,`id`, `description` for storing metadata without cluttering the main content. JSON and YAML do not natively support attributes, requiring metadata to be part of the main structure, reducing clarity) |
| **Extensibility** | High | Medium | Medium | XML is highly extensible with custom tags, allowing easy adaptation to new requirements in safety cases. JSON and YAML are also extensible but lack the rich tooling support XML offers. |
| **Data Integrity & Interoperability** | Strong | Medium | Medium | XML enjoys general acceptance in many industries, including automotive and aeronautics, for its focus on integrity of information and interoperability. By contrast, JSON and YAML, with their lesser complexity, can not necessarily deliver an equivalent level of integrity and compatibility in high regulation environments. |

Each element involved in GSN has as been assigned to an equivalent in XML format. In this way, we will be able to provide a textual representation of GSN-based SCs in XML. The developed XML syntax guarantees that each component of the SCs is clearly defined, traceable, and easily interpretable by both machines and humans.

The core elements of GSN (goals, strategies, contexts, solutions, assumptions, justifications, and evidence) are represented using special XML-tags, each identified by unique attributes (e.g., "id" and "description"). The importance of these attributes is to show the role and content of each element. On the other hand, the hierarchical relationships between them ("SupportedBy", and "InContextOf") are captured by nested structures in these XML formats.

▷ **Goal element:**

Listing 4.1: XML Representation of a Goal Element in GSN

```
<goal id="G1" description="Collision Avoidance System is
    acceptably safe" type="SupportedBy">
    <!-- Nested elements such as context, strategy, assumptions,
        justifications, and sub-goals go here -->
```

```
</goal>
```

**Explanation:** The "goal" tag presented in this example, represents a high-level safety claim, that the Collision Avoidance System (CAS) is acceptably safe. We can also see two others attributes in the XML format. The "id" attribute uniquely identifies this goal node within the SCs. The "description" attribute provides a concise statement of the claim..

▷ **Context element**

Listing 4.2: XML Representation of a Context Element in GSN

```
<Context id="C1" description="System overview: uses sensors and AI
    to detect and prevent collisions" type="InContextOf"/>
```

**Explanation:** Here, the "context" tag essential background to the goal (e.g. the technology used by system such as : sensors and AI). The "type" attribute, "InContextOf", is new. It simply indicates that the context node is related to the goal node by providing further clarity on the system's operation.

▷ **Strategy element**

Listing 4.3: XML Representation of a Strategy Element in GSN

```
<strategy id="S1" description="Ensure safe vehicle operation" type
   ="SupportedBy">
    <!-- Nested elements such as sub-goals, contexts, and
        assumptions go here -->
</strategy>
```

**Explanation:** In this example, the "strategy" outlines the general approach to demonstrate that the vehicle operates safely, which will be supported by more specific goals and evidence.

▷ **Assumption element**

Listing 4.4: XML Representation of a Assumption Element in GSN

```

<assumption id="A1" description="Sensors function correctly" type=
    "InContextOf"/>
```

**Explanation:** The "assumption" here indicates a key assumption that the sensors used in the system are functioning correctly, a condition on which the safety claim depends.

▷ **Justification element**

Listing 4.5: XML Representation of a Justification Element in GSN

```
<justification id="J1" description="Sensor accuracy verified in
    testing" type="InContextOf"/>
```

**Explanation:** The "justification" tag explains why the assumption about sensor accuracy is reasonably, referencing testing that has been conducted.

▷ **Solution element**

Listing 4.6: XML Representation of a Solution Element in GSN

```
<solution id="E1" description="Sensor testing documentation" type=
    "SupportedBy"/>
```

**Explanation:** The "solution" tag points to a specifies piece of evidence, such as documentation of sensor testing, that supports the goal or strategy.

The following is a complete template of a SCs written in XML, structured according to the GSN notation:

Listing 4.7: Comprehensive XML representation of a SC using GSN

```

<goal id="G1" description="Collision Avoidance System is acceptably
    safe" type="SupportedBy">
    <context id="C1" description="System overview: Uses sensors and AI
        to detect and prevent collisions" type="InContextOf"/>
    <context id="C2" description="Boundaries include LIDAR, radar,
        cameras, and AI algorithms" type="InContextOf"/>
    <strategy id="S1" description="Ensure safe vehicle operation" type=
        "SupportedBy">
        <goal id="G2" description="Detect obstacles accurately"  type="
            SupportedBy">
            <context id="C3" description="Sensor subsystem detects
                obstacles" type="InContextOf"/>
            <assumption id="A1" description="Sensors function correctly
                " type="InContextOf"/>
            <justification id="J1" description="Sensor accuracy
                verified in testing" type="InContextOf"/>
            <solution id="E1" description="Sensor testing documentation
                " type="SupportedBy"/>
        </goal>
        <goal id="G3" description="Perform emergency braking timely"
            type="SupportedBy">
            <context id="C4" description="Braking system prevents
                collisions" type="InContextOf"/>
```

```
            <assumption id="A2" description="Braking response time
                meets requirements" type="InContextOf"/>
            <justification id="J2" description="Braking response
                validated in simulations" type="InContextOf"/>
            <solution id="E2" description="Braking system test results"
                 type="SupportedBy"/>
        </goal>
        <goal id="G4" description="Provide driver alerts"  type="
            SupportedBy">
            <context id="C5" description="Alerts warn driver of
                potential collisions" type="InContextOf"/>
            <assumption id="A3" description="Alerts function within
                specified timeframes" type="InContextOf"/>
            <justification id="J3" description="Alert timing validated
                in testing" type="InContextOf"/>
            <solution id="E2" description="Alert system validation
                records" type="SupportedBy"/>
        </goal>
    </strategy>
</goal>
```

### 4.2.1 *Prompt Engineering*

**Prompt Engineering** is relatively new and it is an essential area of NLP [13], especially with the arrival of LLMs. Prompt Engineering focuses on the design and development of prompts that can guide LLMs in an effective way to produce accurate and contextually appropriate outputs. The use of LLMs in a wide range of applications is increasing (from content generation to technical documentation). This evolution shows that well-structured prompts are very important nowadays. A well-structured prompt acts like a bridge between the user's goals and the model's output, by guiding the model to produce a response that closely aligns with the desired result.

In the case of LLMs, two general types of prompts have been most prominently distinguished: system prompts and user prompts.

▷ **User prompt** forms the input entered by the user, and can range in form, for instance, a question, a direction, or a request, to which the AI-system then responds. User prompt is a catalyst for conversation or AI interaction, and guides the following activity or reaction.

  **Example:**

```
User Prompt: "Explain how photosynthesis works"
AI Response:      "Photosynthesis is a biochemical process utilized by
green plants and several other living organisms to convert sunlight
into chemical form. This operation takes place through the creation
of nutrients out of carbon dioxide and water. This mechanism involves
chlorophyll and yields a secondary output =, namely, oxygen."
```

In the mentioned example, it can be seen that the request of a user correspond to a specific query towards an explanation of photosynthesis. The AI then creates a pertinent output in relation to such a query.

▷ **A system prompt**, in contrast, forms an integral guideline or configuration that controls the behavior or tonal aspects of an AI system. This feature can go unacknowledged for a user and is often determined at development stage or modifiable to mold behavior of a system with regard to output. It describes a desired behavior of an AI in terms of conduct (e.g., to respond in a polit, and concise manner) and controls contextual or stylistic aspects of generated output.

**Example 1: (Formal Tone)**

```
System Prompt:       "You are a knowledgeable professor. Be formal and
detailed in your explanations."
User Prompt: "Can you explain how photosynthesis works"
AI Response:     "Photosynthesis constitutes a biochemistric mechanism
through which photosynthesizing plants, algae, and certain types of
bacteria convert light energy, in most cases derived from sunlight,
into a chemical from represented in glucose. This complex mechanism
takes place for the most part in specific organelle structures, namely,
chloroplasts, in which pigments present in chlorophyll trap light
energy in order to enable sugar molecule synthesis."
```

**Example 2: (Causal Tone)**

```
System Prompt:    "You are a friendly guide. Be conversational and keep
things simple."
User Prompt: "Can you explain how photosynthesis works"
AI Response:      "Sure! So, photosynthesis represents a biochemistric
mechanism through which plants produce food for themselves. With
sunlight, plants transform water and carbon dioxide into glucose.
And, at the same time, plants produce oxygen as a byproduct; It can
be compared to a form of "nutrition", taking place in a manner most
predominantly in the leaves of a plant."
```

In the examples above, we notice that, the user's prompt is the same (about photosynthesis). But the system prompt guides how the AI will deliver the information: in a formal, detailed form or informal, simplified one.

That is what makes prompt engineering important, as it enhances the effectiveness of LLMs by optimizing their interaction with users. Although LLMs have been trained on huge datasets [61], they may not naturally understand the specific details and requirements of a task the specific requirements of a task without a human guidance. This is where Prompt Engineering plays a crucial role [47]. In the generation of GSN-based SCs, for instance, if a prompt is clearly defined, it can significantly influence the model's ability to produce the correct XML- structure and content, and therefore the overall utility of the generated output.

To construct an efficient system prompt, several fundamental principles must be considered [44]. First of all, we have to take in consideration the clarity. That means, when we are attempting to write a prompt, we should clearly state what is the task to be performed by the model and in which format the output should be generated. Stating this will reduce the ambiguity that could lead to irrelevant or inaccurate responses. For example, when we request an XML output, specifying the required tags and attributes can help the model to generate the appropriate representation. Additionally, in order to enhance the model's understanding of the model, we could also specify the context with the prompt. This context could contain explanation and definition of specific terms or provide examples of template, which can guide the model toward producing more accurate results.

Another core aspect of good prompt engineering is specificity. When prompts are ambiguous, they tend to produce vague results. On the other hand well written prompts with specific criteria produce more precise answers. If out example we want to generate GSN-based SCs in XML representations. To achieve that, the model can be greatly aided in the output it generates, by specifying the hierarchy of elements and their interrelations. And then there is iteration. After generation of first answer by the model, we are then able to discern aspects where we need to modify the prompt in order to improve the performance.

### 4.2.2 *Configuring Llama-3-1-405B-instruct for Generating the Safety Analysis and the System Design*

The increasing complexity of safety-critical systems demands rigorous safety and design analysis to prevent failures. LLMs present an opportunity to assist in the generation of detailed safety-analyses and system designs [10], [23].

This chapter focuses on configuring Llama-3-1-405B-Instruct, by writing high-level system prompts to generate safety analyses and system designs. It outlines important aspects to take in consideration when it comes to define safety-related parts of a system, to ensure that hazards are identified and mitigated, and ensuring the safe operation of complex systems.

To effectively guide the LLM in generating safety analyses and system descriptions, the system prompt must clearly outline the following key elements: definition of the task the LLM should accomplish, assign it a role in carrying out this task, remind it of the definition of safety analysis and system design, provide it some examples of safety analysis and system design, set the requirements that the result must meet.

All these elements should be taken into account when configuring Llama-3-1-405B-Instruct for generation of the safety analysis and system design .

The different elements of a safety analysis and system description are as follows; system definition and boundaries, the operational environment, functional and performance requirements, safety-critical Function, system architecture and design, implementation details, verification and validation, regulatory and compliance information, maintenance and operation, stakeholders involvement, hazard analysis, risk assessment, and risks mitigation strategies. These elements are further detailed below:

1. **System Definition and Boundaries**

   **System overview:** The LLM should clearly defines the system's purpose, scope, and functions (including its interactions with the environment and the other systems). Explain the purposes to which the system is designed and its basic operations.

   **Boundaries:** The LLM should clearly define the system's boundaries, including its interfaces with other systems and external entities. The LLM should also try to specify as clear as possible, what is included within the system and outside.

**Subsystems and Components:** The LLM should break down the system into its major subsystems and components. Describe how these subsystems and components interact with each other.

2. **Operational Environment**

   **Physical Environment:** The LLM should mention any environmental constraints or requirement and also describe in which conditions the system operates (temperature, humidity, vibration, etc.).

   **Operational Scenario:** The LLM should describe typical and extreme scenarios in which the system will be used. It should not forget to include other details (descriptions of normal operations, maintenance activities, emergency situations, etc.).

   **Human interaction:** The LLM should explain how operators, users, and maintenance personnel will interact with the system and also detail any human-machine interface and requirements that the users have to fulfill.

3. **Functional and Performance Requirements**

   **Functional Requirements:** The model should give a detailed description of the system, its intended functions and a detailed inventory of all system functionalities and their expected outcomes.

   **Performance Requirements:** Here, the LLM should specify the performance criteria the system must meet (speed, accuracy, capacity, reliability, etc.). If necessary, the model should also be able to define measurable benchmarks for performance.

4. **Safety-critical Functions**

   **Identification:** The LLM shall identify all functions that are critical to the system's safety. These are usually the functions whose failure could result in accidents or hazardous situations.

   **Failure Modes:** The LLM should be able to describe the possible ways in which safety-critical functions can fail. It should also consider various failure scenarios and their triggers.

   **Consequences of failure:** The LLM shall explain what happens to safety if a safety-critical function fails. It should also discuss what the possible outcomes and severity of such failures are.

5. **System Architecture and Design**

**Architectural Overview:** The model has to show a very good detailed description of the system's architecture. This description includes important components and their interactions.

**Design Principles:** Design principles applied in designing the system should be described; in particular, those related to safety, such as the fail-safe design and the redundancy.

**Safety Mechanisms:** The LLM should describe specific design features incorporated to ensure safety (fault tolerance, error detection mechanisms, etc.).

6. **Implementation details**

**Software and Hardware:** It is critical for the LLM to present a thorough report of software and hardware components involved. It must include information about technology stack and actual hardware involved.

**Integration:** The LLM should clarify how individual parts integrate and interact with each other in the system. It is critical to highlight any interdependencies and interfaces between these parts.

**Configuration Management:** The LLM must detail methodologies adopted to changes in a system. It must include information about version control and change management processes involved.

7. **Verification and Validation**

**Testing Procedures:** The model has to indicate the methodologies and processes used in the assessment of the system. These testing procedures include unit and system-level testing.

**Validation methods:** The model has to state how the system is to be validated against its requirements. It includes methods such as simulations, formal methods, and field testing.

**Test results:** The model has to show the results of the tests with their interpretation in respect to safety. It should also include evidence of compliance with safety requirements.

8. **Maintenance and Operation**

**Maintenance procedures:** The LLM has to describe regular and preventive maintenance procedures required to ensure continued safety. These maintenance procedures include schedules and responsible parties.

**Operational procedures:** The LLM has to provide detailed operating procedures (start-up, shut-down, and emergency procedures, etc.).

**Incident response:** The LLM has to outline procedures responding to incidents and failures. This incident response includes procedures for identifying and resolving problems.

9. **Regulatory and Compliance Information**

   **Applicable standards:** The model has to provide a list of the safety standards and regulations applied to the system : ISO 26262, SOTIF, DO-178C, etc.

   **Compliance strategy:** The model has to provide an overview of how the system is compliant with regulatory requirements and standards. This compliance strategy includes any compliance measures and documentation.

   **Certification:** The LLM has to note if the system has received or requires any certification. This include details on the certification process and current status.

10. **Stakeholder Involvement**

    **Stakeholder Requirements:** LLM should specify specific requirements and expectations of stakeholders (including operators, users, regulators, etc.).

    **Communication:** approaches adopted for communicating information regarding safety-related items to stakeholders should be detailed. Include strategies for ongoing engagement and feedback.

11. **Safety Analysis**

    **Hazard Identification:** The aim here is to identify all potential hazards that may arise during the operation of the system. The focus is on a wide range of sources. A deep knowledge of the operational context of the system, its environment, and the interactions between its components and human operators is important in order to identify all potential hazards. Effective hazard identification lays the foundation for the subsequent steps in risk assessment and mitigation.

    In the setup of LLM to generate hazard identifications, a carefully well-structured prompt can guide the model in considering a range of sources of hazards. These sources typically include:

    **Operational Hazards:** These hazards result from operation of the system, together with from abnormal or failure conditions. For example:

    **System Failures:** Failures in such components as sensors, actuators, or processing units may lead to hazardous situations. In an autonomous Emergency Braking System (AEBS), a failure in the decision-making unit that delays the issuing of the braking command can end in a crash.

**Functional Degradation:** This refers to a gradual loss in system effectiveness that can happen over a long time or under certain conditions. Let's take an example: in an autonomous vehicle, environmental factors like heavy rain or snow might degrade the performance of sensors, leading to reduced perception capabilities.

**Risk Assessment:** This involves evaluating the identified hazards to determine their potential impact and likelihood of occurrence.

**Risks Mitigation Strategies:** These are approaches to reduce or control the identified risks and interactions within the system.

Considering its size, the system prompt for the generation of Safety analyses and system description are detailed in Listing A.1

### 4.2.3 *Configuring Llama-3-1-405B-instruct for Generating Safety Cases*

"Configuring" in this section, refers to the process of writing system prompts that direct the behavior of the Llama-3-1-405B-instruct to, ideally, generate GSN-based SCs in XML format.

In order for the Llama-3-1-405B-instruct model to learn to generate valid GSN-based SCs in XML, these prompts must to effectively guide the LLM in generating accurate SCs, it is important to explicitly state in the system prompt what kind of output is expected and under which constraints. The structure of the prompts includes several key components: Definition of the task to be performed by the LLM and assign it a role in which it must accomplish the task, remind it of the definition of SCs and GSN, remind it of the various key elements making up GSN as well as defining them, show how each of these elements can be written textually in XML, show how each of these elements be put together to constitute a safety case in XML that uses GSN, remind the LLM that safety standards must be taken into account, and set requirements that the output must meet to be acceptable.

After following all these steps, we set up our system prompt for generating safety cases in XML using the GSN.

Considering its size, the system prompt for the generation of Safety Cases in XML using the GSN is detailed in the Appendix A.2.

## 4.3 DATASET PREPARATION

Since we want to fine-tune our tiny Llama3 8B model, we need data (input = safety analysis and system description; output = GSN-based SCs in XML). Since we haven't been able to find a dataset that corresponds to the type of data we want, we'll have to generate them ourselves, using Llama-3.1-405B-Instruct.

In this chapter, we provide a comprehensive overview of the methodology used to generate the dataset essential for streamlining SCs generation in XML format using GSN framework. The dataset has been automatically generated by a small tool (developed in Python) that integrates the Llama-3.1-405B-Instruct model to generate "Safety Analysis and System Design" as well as corresponding "Safety Cases in XML" based on predefined system names and system prompts. Following steps present the process for the generation of the dataset:

1. A list of 1,000 names of safety-critical systems has been generated using ChatGPT.
2. Once the list of 1,000 names has been generated, it is saved to a file.
3. The user presses the "Generate Dataset" button on the tool's user interface.
4. Once the user clicks on "Generate Dataset" button, the model (Llama-3.1-405B-Instruct) will be called via an API.
5. After that, the model through the list of 1000 System names, from first name to last, executing the 2 following last steps.
6. Based on the system name, the model will generate a safety analysis and a system design and then the safety cases in XML using GSN which will be displayed on the user interface.
7. Safety analysis and system design as well as the SCs generated by the model will be saved in an excel file to form the dataset.

The following image shows the entire dataset generation process.

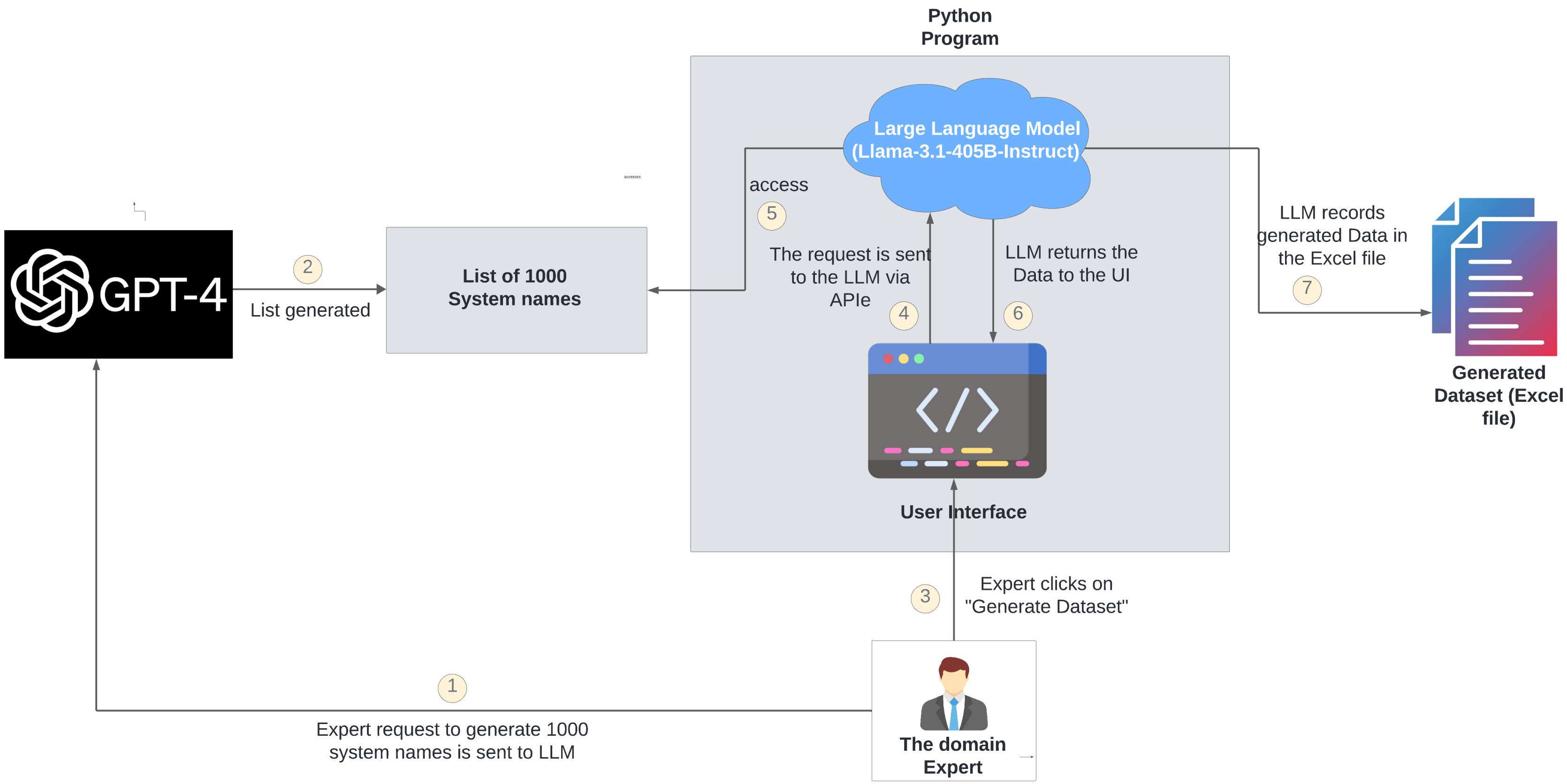


Figure 4.2: Overview of the dataset generation process

The following sections describe the structure of our python program, the format of the dataset and the reasons why Llama-3.1-405B-Instruct was chosen for this task.

### 4.3.1 *Dataset overview*

Our dataset contains around 1,000 rows. In other words, 1,000 examples of safety-critical systems from real-world. Those systems come from different safety domains (aviation, automotive, aero-spacial, etc.). The dataset contains 03 columns:

- ▷ **System Name:** it's unique for each.
- ▷ **Safety Analysis and System Design:** covers various system-related aspects, such as system description and boundaries, operational environment, functional and performance requirements, safety critical functions, hazards analysis, system architecture and design, implementation details, regulatory and compliance information, stakeholders involved.
- ▷ **Safety Cases in XML:** Here, we mean GSN-based SCs structured in XML-format. It has to be noted that these SCs are generated on the basis of the Safety analysis and system description provided as input to the model.

This structured dataset enables the program to serve as both a practical tool for safety engineers and a source of training data for further model developments.

The following image shows the dataset stored in an Excel file.

| | A | B | C |
|---|---|---|---|
| 1 | System Names | Safety Analysis and System Design | Safety Cases in XML |
| 2 | Aircraft Collision Avoidance System | ### System Description and Safety Analysis: Aircraft ( | <goal id="G1" description="Aircraft Collision |
| 3 | Advanced Air Traffic Control System | ### System Description and Safety Analysis: Advance | <goal id="G1" description="The Advanced Ai |
| 4 | Flight Navigation System | ### System Description and Safety Analysis: Flight Na | <goal id="G1" description="The Flight Navig |
| 5 | Engine Monitoring System | ### System Description and Safety Analysis: Engine M | <goal id="G1" description="The Engine Mon |
| 6 | Aircraft Fuel Management System | ### System Description and Safety Analysis: Aircraft F | <goal id="G1" description="Aircraft Fuel Ma |
| 7 | Weather Alert System | ### System Description and Safety Analysis: Weather | <goal id="G1" description="The Weather Ale |
| 8 | Air Traffic Management System | ### System Description and Safety Analysis: Air Traffi | <goal id="G1" description="Air Traffic Mana |
| 9 | Runway Incursion Prevention System | ### System Description and Safety Analysis: Runway | <goal id="G1" description="Runway Incursio |
| 10 | Aircraft Emergency Response System | ### System Description and Safety Analysis: Aircraft E | <goal id="G1" description="Aircraft Emerger |
| 11 | Flight Control System | ### System Description and Safety Analysis: Flight Co | <goal id="G1" description="The Flight Contr |
| 12 | Avionics System | ### System Description and Safety Analysis: Avionics | <goal id="G1" description="The Avionics Sys |
| 13 | Cockpit Display System | ### System Description and Safety Analysis: Cockpit I | <goal id="G1" description="The Cockpit Disp |
| 14 | Navigation and Communication Syst | ### System Description and Safety Analysis: Navigati | <goal id="G1" description="Navigation and ( |
| 15 | Surveillance System | ### System Description and Safety Analysis: Surveilla | <goal id="G1" description="The Surveillance |
| 16 | Airborne Collision Avoidance System | ### System Description and Safety Analysis: Airborne | <goal id="G1" description="Airborne Collisio |
| 17 | Autonomous Drone Navigation Syste | ### System Description and Safety Analysis: Autonon | <goal id="G1" description="Autonomous Dr |
| 18 | Aircraft Terrain Avoidance Warning S | ### System Description and Safety Analysis: Aircraft T | <goal id="G1" description="The Terrain Avoi |
| 19 | Unmanned Aerial Vehicle (UAV) Cont | ### System Description and Safety Analysis: Unmann | <goal id="G1" description="The UAV Contro |

Figure 4.3: Overview of the generated dataset in Excel

### 4.3.2 *Program Structure for automated Dataset Generation*

On the User Interface (UI) of our tool developed in Python, the user has the possibility to start the generation of the dataset by clicking on "Generate Dataset" button. Below, we present the various components of the user interface and description of how the program works:

It is important to remind that the model uses the system prompt defined in the previous section for generating safety analysis and system descriptions, and another for generating GSN-based SCs in XML.

The dataset generation process begins by the established list of 1,000 system names (generated by ChatGPT), which is saved in a file. These system names form the foundation for generating corresponding safety analysis and SCs. In the image above, we can see the system name displayed. This name comes from the list of 1,000 names previously established. Directly below left, we have the safety analysis and system description generated from the system name. On the far right, we have the safety cases generated from the content on the left. Once these data are generated, the system saves the data in an excel file and then moves on to the next system name and performs the same generation. This step is repeated for every 1000 name systems. At the very bottom, we have a gauge showing the generation progress.

The output is organized into three columns **("System Name", "Safety Analysis and System Design", "Safety Cases in XML")**. These three elements form the core of the dataset, which is saved for further use in automating SCs generation. It is important to remind that the model use the system prompt defined in the previous section for generating safety analysis and system descriptions. Based on the generated safety analysis and system description, the model will also use the defined system prompts to generate the GSN-based SCs in XML-format.

### 4.3.3 *Justification for Using Llama-3.1-405B-Instruct*

The Llama-3.1-405B-Instruct model was chosen for the generation of our dataset of 1,000 system names, because it is one of the largest open-source LLMs available. Llama-3.1-405B-Instruct can generate complex and high-quality GSN-based SCs in XML-format. This model has been trained on large diverse datasets, which enables it to handle complex tasks such as generation of safety analysis, system description and SCs in XML format.

The capabilities of this model to work with system prompts, as described in the previous chapter, also enables a streamlined approach to generating consistent, high-quality safety data across a large dataset.

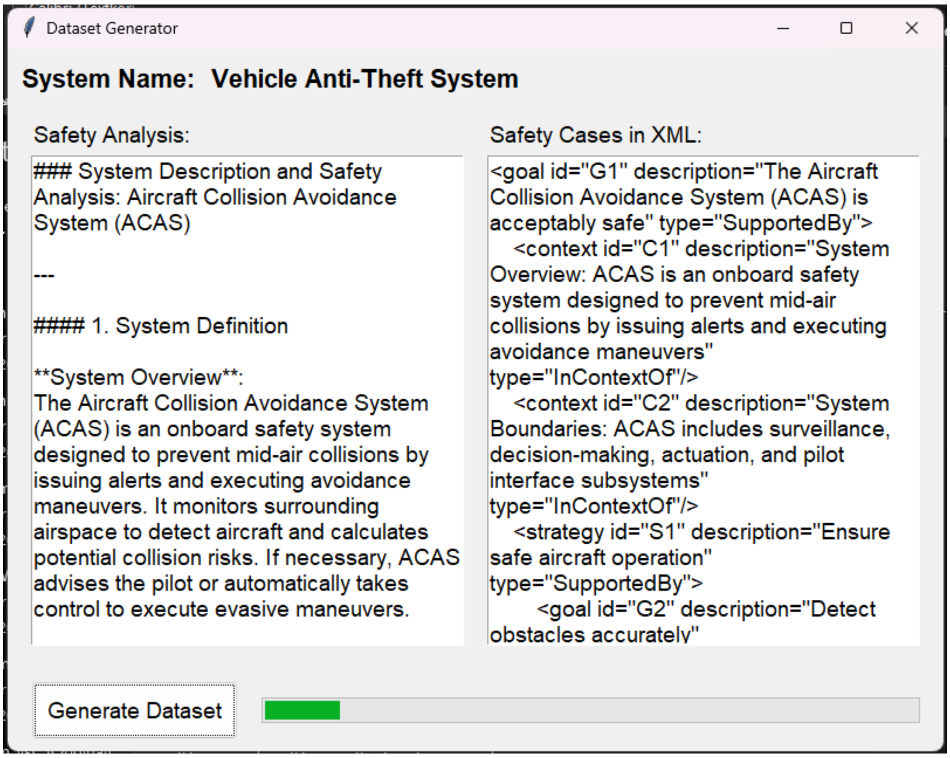


Figure 4.4: User Interface of the program for generating the dataset automatically

### 4.3.4 *API Integration and Code*

The Llama-3.1-405B-Instruct model is accessed through an API of NVIDIA NIM (NeMo Inference Microservices) integrated into the dataset generation program. NVIDIA NIM [1] is an important framework designed for deploying large AI models in a variety of environments. It allows IT-teams to self-host LLMs like Llama-3.1-405B-Instruct in their own managed infrastructures. This is particularly useful for organizations wo want to integrate LLMs into their applications without relying on third-party cloud providers. With NIM, those organizations have the possibility to get an APIs (Application Programming Interface) and tools that will make it easier for developers to build and deploy generative AI models (chatbots, assistants and more, at scale). The following code snippet illustrates how the API is used to query the model and generate the safety data:

## 4.4 API CODE EXAMPLES

Below is an example of the API code used in the project to generate the safety analysis and the system design based on the system name.

1 https://www.nvidia.com/en-us/ai/

It is important to note that a template of this API code is provided on the Nvidia NIM website [2]. We have used this template and adapted it to our specific needs [46]

The API code for Running Llama-3.1-405B-Instruct for safety analyses and system description generation in XML is detailed in Appendix A.9.

Below is an example of the API code used in the project to generate SCs in XML based on safety analysis and the safety design previously generated. The code is detailed in appendix A.10

Once the dataset has been generated, we'll publish it.

### 4.4.1 *Publishing the generated Dataset on Hugging Face Hub*

Huggingface, founded in 2016 by Clément Delangue, Julien Chaumond, and Thomas Wolf in New-York City, the company was initially focused on the development of developing a chatbot application for teenager. But the platform has since evolved into a leading hub for machine learning models, particularly those related to natural language understanding and generation. Hugging Face is well-known today in the area of AI, ML and DP for its Transformers Library, which we can find or access to a wide array of pre-trained models that can be used for various tasks (classification, translation, summarization). This name "Hugging Face" derives from the emoji that represents a hug, a symbol for how the organization cares about the community efforts and collaboration in AI development. Most of the existing barriers for those researchers and developers who want to use state-of-art machine learning solutions have been significantly reduced by this library [33]. The platform fosters an open-source community, and encourages the collaboration and the sharing among developers and researchers. Users have the possibility not only to utilize existing models (developed and published by other users) but also to contribute by publishing their own, which enhances the diversity and richness of available resources. On the platform Hugging face, there is also a Model Hub, where users have the possibility to find, deploy models, and the Datasets library, which provides access to numerous datasets for training and evaluation purposes. Moreover, Huggingface emphasizes the importance of ethical AI development, providing resources and guidelines to help users navigate the complexities of deploying AI responsibly. Nowadays, Hugging face is a powerful tool for students, research and for the development of applications in industry. For more information, please visit their website[3]. Our final dataset, after generation, was published on Huggingface[4]. By Publishing the dataset on Hugging Face, we are making it accessible for other researchers and developers of the platform.

2 https://www.nvidia.com/en-us/ai/
3 https://huggingface.co/
4 https://huggingface.co/datasets/Max491/New_Version_XML_Safety_Cases_Archive

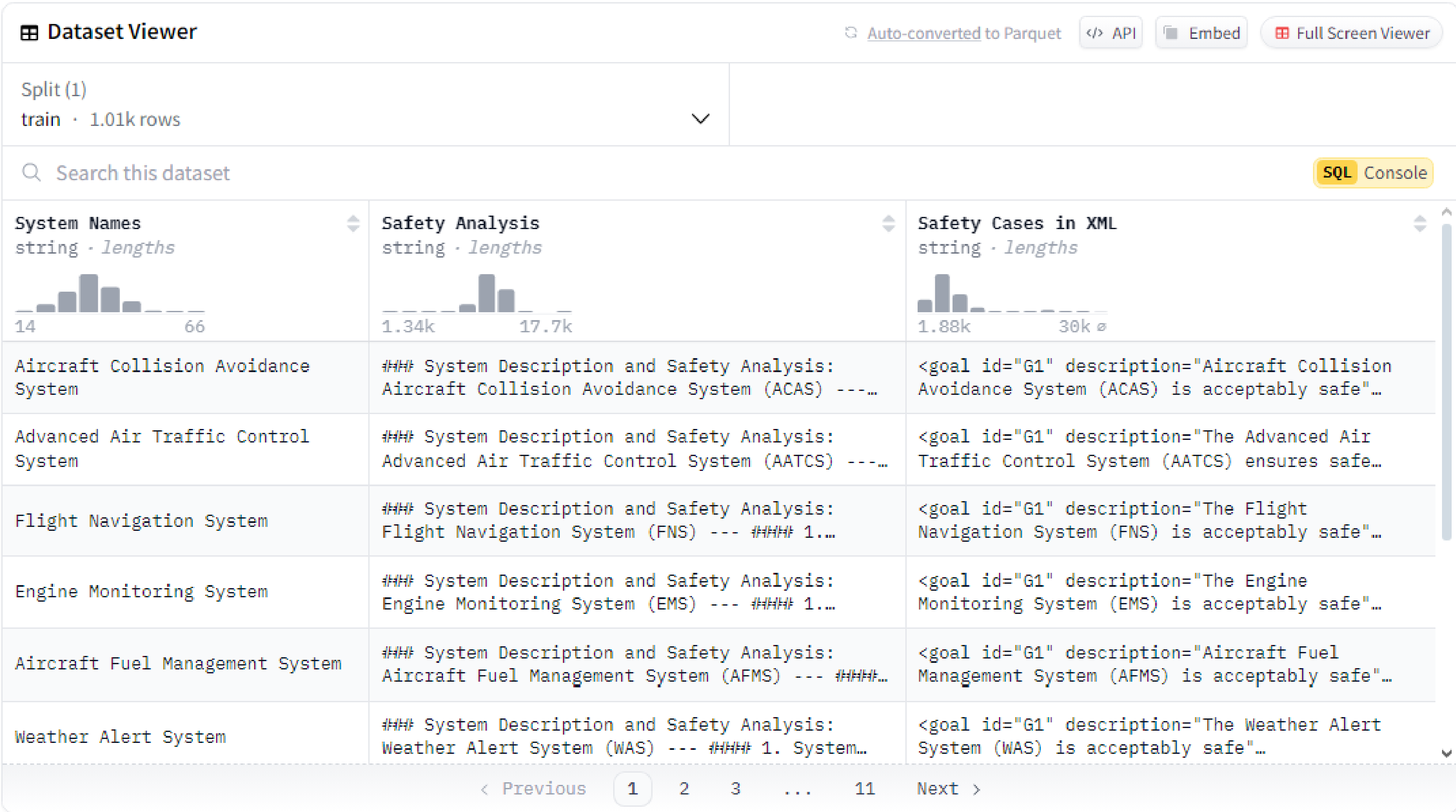


Figure 4.5: Overview of the dataset on Huggingface Hub [5]

## 4.5 MODEL SELECTION AND FINE-TUNING

This section presents the fine-tuning process of our model. We used for the fine-tuning the framework called Unsloth. In the first section, we will explain why we chose this framework.

### 4.5.1 *Introduction of the Framework & Platform Unsloth*

created by Daniel and Michael Han, Unsloth is a platform and framework that helps to facilitate and speed-up LLM's fine-tuning process. This platform is a game changer in the field of AI because its particularity is the fact that users have the ability to accelerate training speeds by up to 30 times while reducing the memory usage by 60% in comparison to traditional methods. This framework makes the training of LLMs very accessible for beginners, democratizing the technology. For more information, you can visit their official website [6]

### 4.5.2 *Fine-Tuning Process using the Unsloth framework*

The fine-tuning process for the selected LLM (Llama3 8B), is essential for transforming its general capabilities into a specified tool aimed at generating GSN-based SCs in XML format.

As a first step, Llama 3.1 (8B) model was downloaded from Hugging Face Hub by using an optimized configuration to efficiently address computational issues. In order to reduce the high memory and computational demand, that are associated with our model, we have used in this work a quantization strategy. That is the reason why, our model was initialized using a configuration called BitsAndBytes (BnB). With this configuration we are able to store the model at a 4-bit precision level (load-in-4bit=True). BnB (Dettmers, Pagnoni, Holtzman, and Zettlemoyer[21]) serves as a streamlined wrapper for specialized Compute Unified Device Architecture (CUDA) operations, facilitating the model compression for efficient storage and memory usage (storage in 4-bit precision) [41].

In order to improve the Parameter-Efficient Fine-Tuning (PEFT), we have followed an approach called the Quantized Low-Rank Adaptation (QLoRA) (Dettmers, Pagnoni, Holtzman, and Zettlemoyer[21]). With this approach, we are tuning just a limited number of parameters, primarily those in critical layers that significantly influence task performance. For Llama 3 (8B), the targeted layers include: up-proj", "gate-proj", "v-proj", "k-proj","down-proj", "q-proj", and "op-roj".

6 https://unsloth.ai/

The maximum sequence length for input data was set to 65536 tokens to accommodate extensive context (Safety analysis and system description + GSN-based SCs) during training.

In the optimization phase of the model, we succeeded to find a good tradeoff between computational efficiency and an improvement in performance of the model by tuning parameters for training with much fastidiousness. A crucial aspect of this optimization process involved the implementation of a conservative learning rate, set at 2e-6 in conjunction with the deployment of the paged-adamw-32bit optimizer (Loshchilov, Hutter, et al. [40]), we have selected one of the variant because it has better efficiency on 32-bit GPU architectures. To navigate hardware constraints while maximizing training efficacy, a moderate batch size was employed in tandem with a four step gradient accumulation strategy, facilitating optimal resource utilization. With 16-bit precision training, activated by setting fp16 to True, played a great role by maximizing the resources of a GPU, while while maintaining the stability of the model during the training. It simultaneously had the dual benefits of potentially halving memory consumption and associated costs, while simultaneous providing the possibility of doubling the training speed.

The model fine-tuning process involved two-epochs of training. This is standard for Large Language Models. This is due to their pre-existing knowledge base and sophisticated architecture design of this LLMs, which facilitate rapid adaptation to new tasks with little retraining duration. Concerning the training configuration, we used a modest batch size of two samples per computational unit, and four-step gradient accumulation. We made these configuration choices to reach a perfect balance in the training dynamics, thus emulating the effect of a bigger batch size while respecting hardware constraints. Moreover, the introduction of gradient check-pointing techniques, to fit larger networks into memory [7], has allowed to optimize the memory utilization. By doing this, we save intermediate computational states, what significantly reduce the overall memory footprint of the training process.

The SFTTrainer class provided by the Transformer Reinforcement Learning (TRL) library [8] has been used for the fine-tuning. This toolkit, seamlessly integrated with the transformers ecosystem, offers a whole suite of instruments for improving transformer-based language models by multiple advanced techniques, including Reinforcement techniques. The training configuration was kept as previously setup with the dataset and tokenizer loaded from Hugging Face Hub.

To make it easier to use Unsloth for fine-tuning LLMs, it's important to note that Unsloth has published ready-to-use Google colab notebooks on their Hugging

7 https://medium.com/tensorflow/fitting-larger-networks-into-memory-583e3c758ff9
8 https://huggingface.co/docs/trl/index

Face profile [9] . We therefore used the template corresponding to the fine-tuning of Llama 3 8B and made the necessary changes to suit our needs in order to achieve our objectives. Below is the source code used to fine-tune our Llama-3.1-8B model [60].

#### CODE-SOURCE FOR FINE-TUNING [60]

The code for fine-tuning our model (Llama-3-8B) is detailed in Appendix A.10.

### 4.5.3 *Deployment of the model on Huggingface*

Once the training is complete, we will be able to save the LoRA and the tokenizer locally. But we can also push the LoRA and the tokenizer to the Hugging Face Hub so that we can use the model anytime, anywhere. To do this, we create a new directory with the model name.

#### CODE-SOURCE FOR SAVING AND PUSHING THE LORA AND TOKENIZER ON HUGGING FACE HUB

The code-source for saving and pushing the LoRA and Tokenizer on Hugging Fcae Hub is detailled in Appendix A.12.

Below we can also see the content of the `"Max491/Safety_Cases_Generator_new"` repository that has been published on HF Hub.

9 https://huggingface.co/unsloth

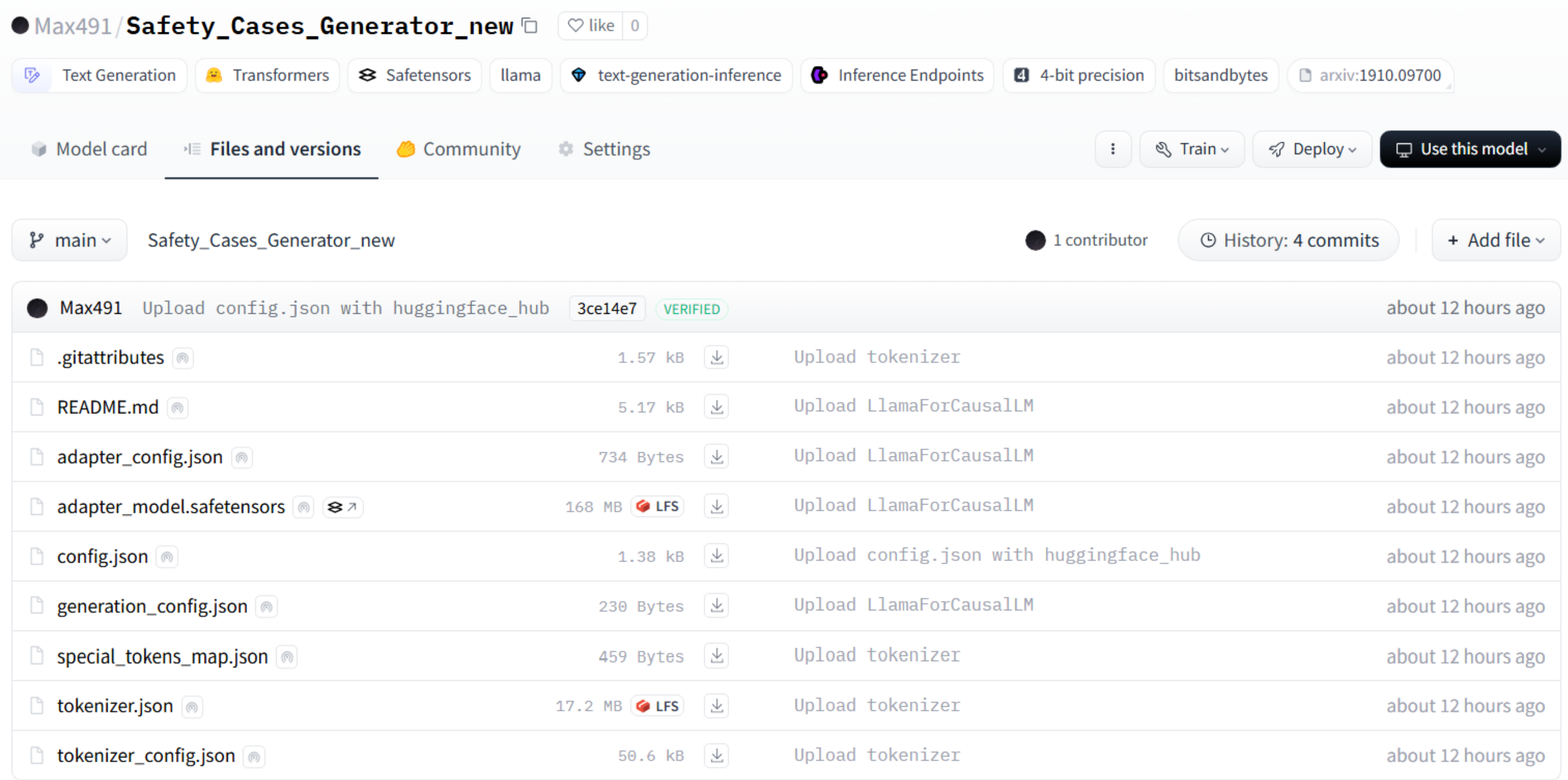


Figure 4.6: Model published on HF [10]

## 4.6 development of the desktop application

In this chapter, we will present the application that we have developed. This application integrates our trained model, for generating GSN diagrams and GSN-based SCs in XML-format. As input, the model takes the safety analysis + system design provided by the user in oder to generate those SCs in XML format. Here, we will discuss the design and architecture of the application, and then detail how its modular structure supports the integration of its advanced functionalities. After that, we will show how we ensure the conversion of GSN-based SCs in XML into GSN diagrams, while highlighting the methodologies we have employed to ensure accurate representation of SCs. Finally, we will examine the GUIs, and present their features.

### 4.6.1 *Design and Architecture*

We developed the application by using the programming language called python and the GUI has been developed using the framework called Tkinter[11]. Our application integrates our fine-tuned LLM to generate GSN-based SCs based on the safety analysis + system design provided by the user.

The main interface consists of a single input field where users can enter their safety analysis and the system description. A prominent button "Generate GSN Diagram" allows users to trigger the generation of SCs, which subsequently leads to the creation of GSN diagrams.

11 https://docs.python.org/3/library/tkinter.html

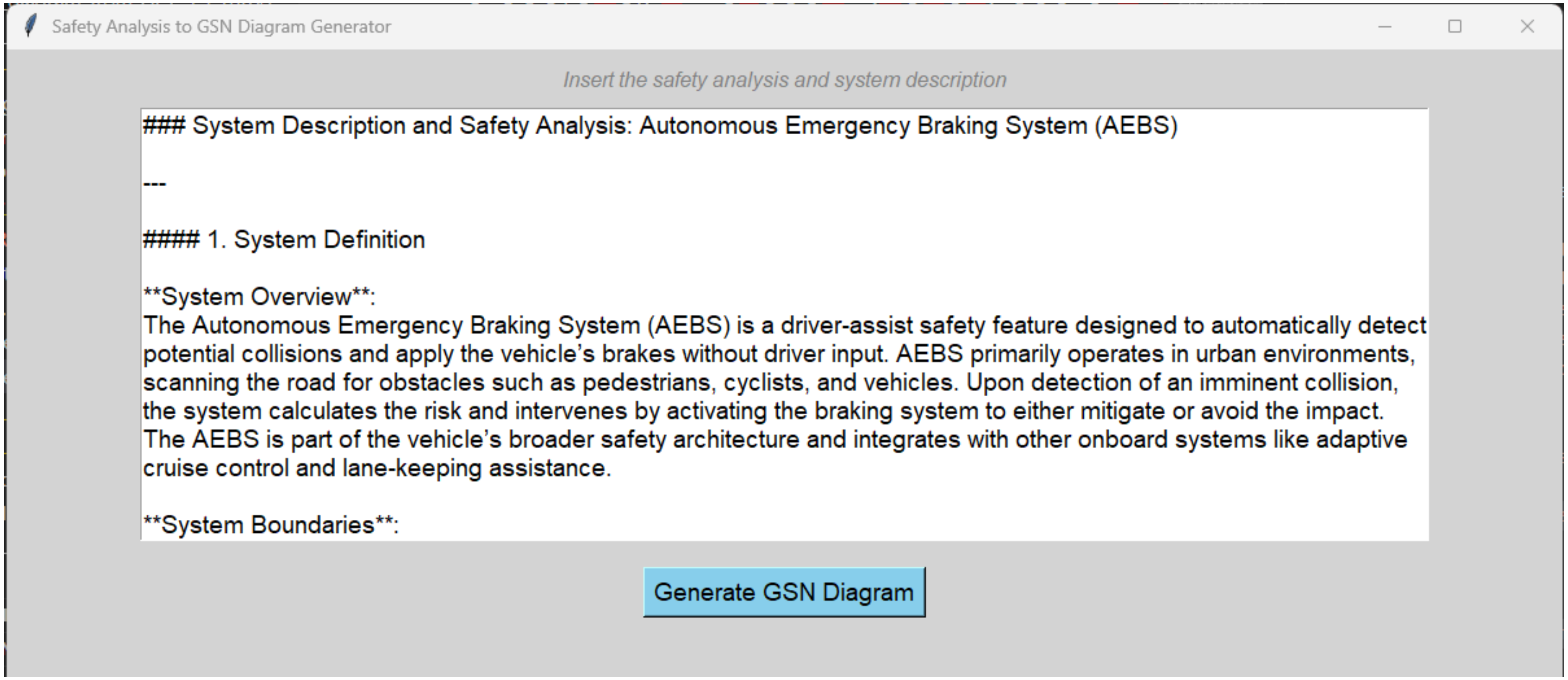


Figure 4.7: Application Homepage

The generated GSN diagrams are displayed in a separate window that is scrollable and re-sizable, allowing users to view large diagrams without losing clarity. Additionally, we see three buttons at the bottom of this window. On the left, a button for viewing SCs in XML. In the middle, we have the button for regenerating the GSN diagram, On the right the button for downloading the GSN-Diagram. Below, we have the picture of this Interface.

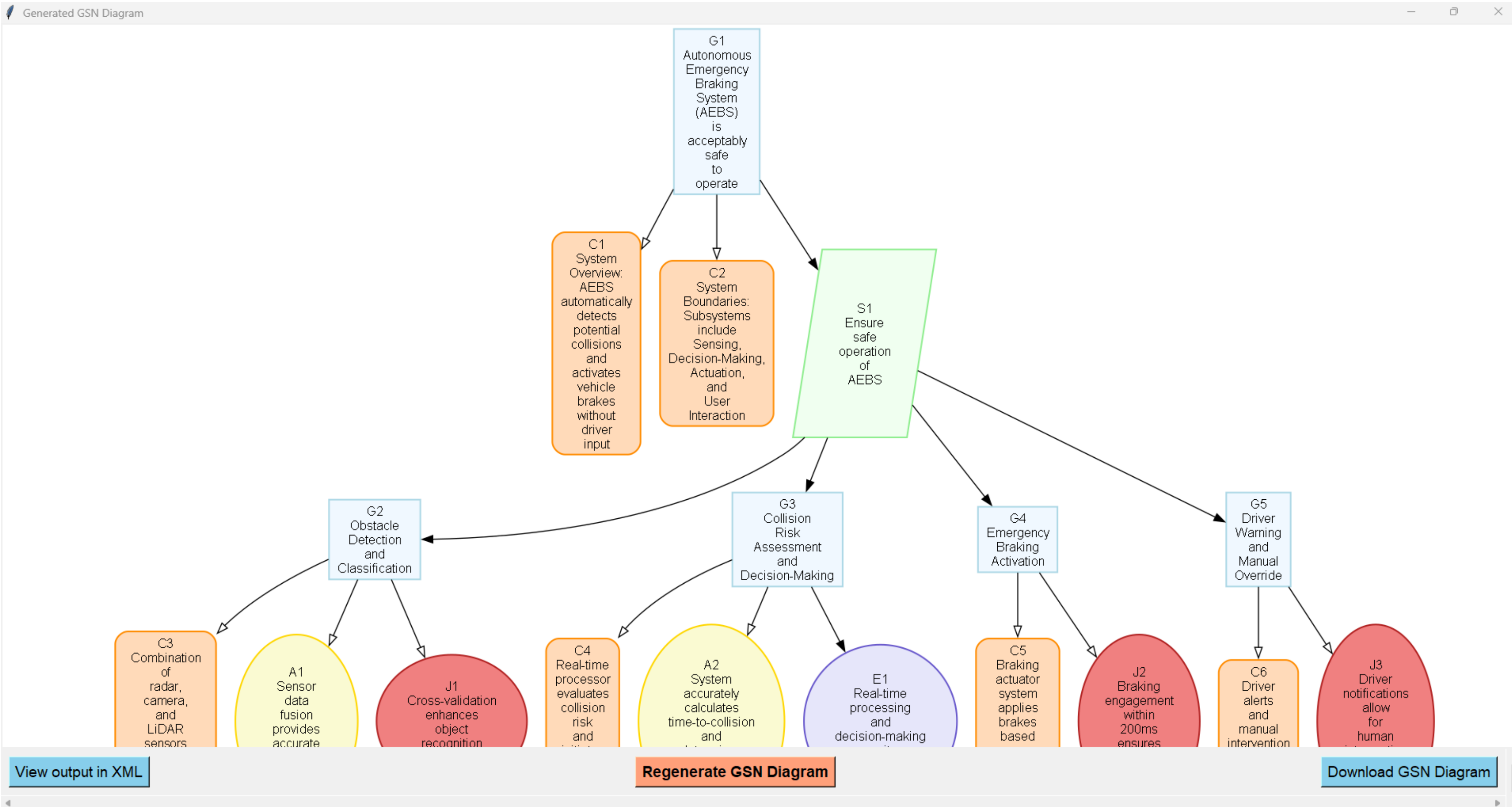


Figure 4.8: Generated GSN-Diagram

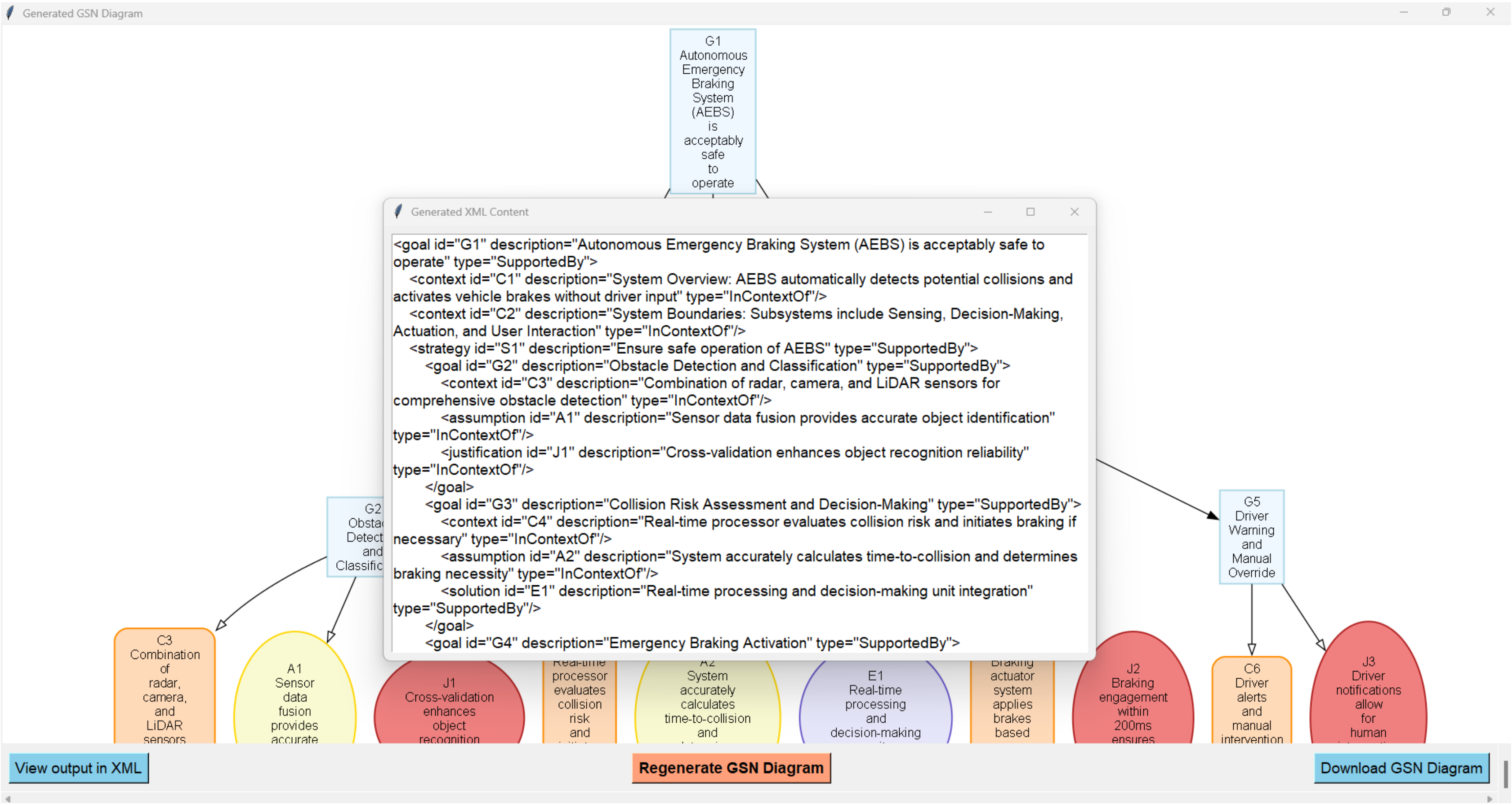


Figure 4.9: SCs in XML

### 4.6.2 *Implementation of XML to GSN Conversion*

One of the most important functionality of our application is the ability to convert XML content into GSN diagrams. The conversion from XML to GSN diagrams is one of the critical feature for our tool. After our tool receives the SCs in XML format, it first employs the "xml.etree.ElementTree" module to parse the XML content and extracts the useful information. The parser scans the XML content and constructs a tree structure where each node represents an XML element. Each element contains attributes, nested elements and text. After parsing, each XML element type (goal, strategy, context, solution, etc.) is mapped to its corresponding GSN component. Each node has a unique identifier ("id" attribut in XML) and a description, both of which are essential for building the GSN hierarchy and labeling each node. GSN uses relations, such as "SupportedBy" or "InContextOf", to represent relationships, which are necessary to structure safety arguments. An XML parser traverses recursively the XML tree and constructs relationships between nodes depending on thier hierarchical level and attributes. Once the nodes and relationships have been structured, the program will convert this setup into a GSN diagram using the graph visualization library called Graphviz [12].

The approach that we present in this chapter provides a systematic approach to leveraging LLMs for SCs generation. Later, in chapter 5, a practical case study on the U-SHIFT autonomous vehicle is detailed to test the proposed framework, in order to identify its effectiveness and limitations.

12 https://graphviz.org/

# 5 TESTING THE DEVELOPED MODEL ON THE U-SHIFT AUTONOMOUS VEHICLE

In this chapter, we will test the capability of our trained model in generating accurate and structured GSN-based SCs in XML-format. We will test our model against a concrete example from the real-world: the U-Shift vehicle [13]. The German Aerospace Center (DLR) has developed the U-Shift autonomous vehicle. It pursues a completely new approach concerning autonomous, modular transportation that can be integrated into cities. The U-SHIFT vehicle has two main parts consisting of a "driveboard" and interchangeable "capsules". The "driveboard" contains many elements (vehicle's propulsion, steering, braking, and control systems). This part is responsible for the autonomous transport of different capsules. Below, we have an image of the U-Shift vehicle.

## 5.1 PRESENTATION OF THE TESTING STRATEGY

### 5.1.1 *What do we want to test ?*

**Aim:** We want to test if our trained model can generate SCs that consider almost all the relevant details provided in the safety analysis and system description. To achieve this objective, we have identified 5 safety-critical subsystems within the U-Shift autonomous Vehicle that are important to its operation , and on which we are going to carry out tests.

### 5.1.2 *Identified relevant safety-critical sub-systems within the U-Shift Vehicle*

Within the U-Shift, we have identified 5 very important safety-critical systems on which we will be testing.

- ▷ **U-Shift Drive System:** in charge of controlling the movement of the U-Shift vehicle (acceleration, braking, steering, etc.). It is integrated with automated and remote driving technologies, allowing the vehicle to navigate pre-defined routes in urban environments safety. The system interacts with capsule, control system, and other subsystems to ensure smooth operation

13 https://www.dlr.de/en/research-and-transfer/projects-and-missions/u-shift

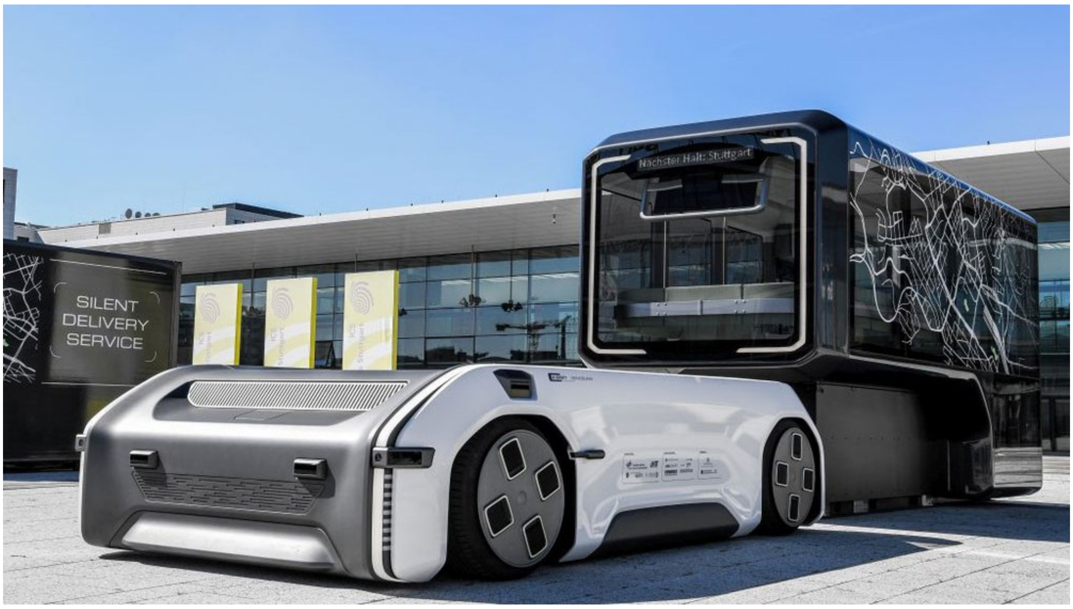


Figure 5.1: U-SHIFT autonomous Vehicle (Credit: DLR (CC BY-NC-ND 3.0))

▷ **U-Shift Onboarding System:** manages many functions (passenger access, including ramp deployment, door operations, interaction with passenger control inputs, etc.). This system operates autonomously, ensuring smooth and barrier-free access to the capsule of all passengers, including those with mobility limitations. It is integrated with the vehicle's Control System to ensure safe and efficient operations during passenger boarding and disembarking.

▷ **U-Shift Control System:** This system represents the central decision-making unit of the U-SHIFT vehicle. It should process sensor data, and manage communication with other subsystems (Drive, Onboarding, Communication, and Emergency systems). It also provides real-time commands. The U-Shift control system guarantees that the vehicle operates in a well-coordinated way concerning navigation, safety and emergency handling for passengers, greatly enhancing the safety of both the autonomous vehicle and the entire system.

▷ **U-Shift Communication System:** It is one of the most important subsystems in the U-Shift vehicle. It should enable the communication between the vehicle itself, external operators, and the passengers. The communication system is supposed to process data concerning the operational status, emergency alerts, and passenger notifications. The importance of this system is that it ensures safe and transparent operation of the vehicle under all driving conditions and, if needed, it provides remote control and monitoring.

- **U-Shift Emergency System:** this subsystem of the U-Shift vehicle is designed to manage critical situations by providing immediate responses (Emergency call, Emergency Stop, system-wide alerts, etc.). This system is very important because it ensures the safety of the passenger and the vehicle, and allow rapid interventions during hazardous conditions and maintains communications with external operators for support.

### 5.1.3 *Testing strategy*

Our testing strategy is simple:

- We have at our disposal a document entitled 'ModelBasedSafetyAnalysisExample', which here represents the 'Safety analysis and system design' of our U-SHIFT Vehicle. Given that we want to generate SCs and system design for each of the systems identified, it is important to firstly generate a safety analysis and system design for each of the systems.

- In order to achieve this, we will use ChatGPT for its capability to understand and interpret documents and analyze diagrams. As such, we will submit to it the document including the safety analysis and system description for the whole of the U-SHIFT Vehicle, and a sample case of a safety analysis, in order for it to generate the safety analysis and system descriptions for each of the subsystems according to the format utilized in model training.

- Once the generation of safety analysis and system design has been completed, it will be submitted to our trained model, which will generate safety cases. Once the safety cases for each of the systems have been generated, we will examine whether the SCs include all the various relevant details present in the safety analysis and system description.

## 5.2 CASE 1: U-SHIFT ONBOARDING SYSTEM

### 5.2.1 *Safety analysis and system description of the U-Shift Onboarding System*

The safety analysis and system description for the U-Shift Onboarding System are shown in appendix A.3.

### 5.2.2 *Generated safety cases of the U-Shift Onboarding System*

The U-SHIFT Onboarding System's safety analysis and system design was submitted to the trained model and the SCs generated by our trained model is detailed in appendix A.4.

### 5.2.3 *Analysis of the U-SHIFT Onboarding System's safety cases in XML*

Table 5.1: Analysis of the U-SHIFT Onboarding System's safety cases in XML

| Test ID | Test Objective | Description | Expected Output | actual XML output | Passed / Failed |
|---|---|---|---|---|---|
| T1 | Top-goal Verification | Ensure the XML includes the top-level goal for the overall safety of the U-SHIFT Onboarding System | We expect to get something whose meaning is close to: "U-SHIFT Onboarding system is acceptably safe", with child contexts, assumptions, justifications, and strategies | We got: Goal G1 with linked contexts (C1, C2), assumption (A1), justification (J1), and strategy (S1) | Passed |
| T2 | Context Verification | Verify the inclusion of system-level contexts, such as system overview and boundaries | We expect information about system boundaries (definition), the subsystems and components, in accordance with the provided safety analysis and system design. | The context nodes C1 and C2 are included and accurately describe the system overview and boundaries. | Passed |
| T3 | Subsystem Decomposition | Validate that XML includes goals for each subsystem (Ramp Control, Door Operation, Obstacle Detection, and Passenger Safety Interface) with proper child elements to support overall safety. | We expect to get something whose meaning is close to: "Ramp Control Subsystem is acceptably safe"; "Door Operation Subsystem is acceptably safe"; "Obstacle Detection Subsystem is acceptably safe" and "Passenger Safety Interface Subsystem is acceptably safe", with child contexts, assumptions, justifications, and strategies | Sub-goals (G2, G5, G8, and G11) are present with appropriate child elements. | Passed |

| Test ID | Test Objective | Description | Expected Output | actual XML output | Passed / Failed |
|---|---|---|---|---|---|
| T4 | Hazard Identification and Mitigation | Verify that hazards associated with the Onboarding System are identified and mitigations are represented in the safety case. | The XML should include goals addressing specific hazards (Ramp not retracted, unintented door opening, obstacle not detected, delay in ramp or door operation). Each hazard should have a clear mitigation strategy and solutions to demonstrate its handling. | Sub-goals (G3, G4, G6, G7, G9, G10, G12 and G13) address perfectly the identified Hazards. Mitigation Strategies (S2, S3, S4) and Solutions (E1, E2, E3, E4, E5, E6, E7, E8) are properly linked. | Passed |
| T5 | Regulatory Compliance Mapping | Ensure that the safety case includes contexts, assumptions or justifications referencing compliance with regulatory standards | We expect the following standards: ISO 26262, UNECE R107, SAE J3016 provided in the safety analysis and system design | Nodes (J1, A1, A2, A3, A4, and A5) mention standards but do not explicitly state which of the standards they refer to among: ISO 26262, UNECE R107, J3016. | Test partially passed |

## 5.3 case 2: u-shift control system

### 5.3.1 *Safety analysis and system description of the U-SHIFT Control System*

The safety analysis and system description for the U-SHIFT Control System is shown in appendix A.5.

### 5.3.2 *Generated safety cases of the U-Shift Control System*

The U-SHIFT Control System's safety analysis and system design was submitted to the trained model and the SCs generated by our trained model is shown in appendix A.6.

### 5.3.3 *Analysis of the U-SHIFT Control System's safety cases in XML*

Table 5.2: Analysis of the U-SHIFT Control System's safety cases in XML

| Test ID | Test Objective | Description | Expected Output | actual XML output | Passed / Failed |
|---|---|---|---|---|---|
| T1 | Top-goal Verification | Ensure that the top goal defines the overall safety of the U-SHIFT Control System | We expect to get something whose meaning is close to: "U-SHIFT Control system is acceptably safe", with child contexts, assumptions, justifications, and strategies | We got exactly: Goal G1 with linked contexts (C1, C2) and strategies (S1, S2) | Passed |
| T2 | Context Verification | Verify the inclusion of system-level contexts describing the system overview, boundaries and key subsystems | We expect information about system boundaries (definition), the subsystems and components, in accordance with the provided safety analysis and system design. | The context nodes C1 and C2 are included and describe the system overview and boundaries as specified in the safety analysis and system description. | Passed |
| T3 | Subsystem Decomposition | Validate that XML includes goals for Data Processing, Decision-Making, Command Interface, and Diagnostics | We expect to get something whose meaning is close to: "Data Processing subsystem is acceptably safe"; "Command Interface Subsystem is acceptably safe"; "Decision-Making Subsystem is acceptably safe" and "Diagnostics Subsystem is acceptably safe", with child contexts, assumptions, justifications, and strategies | Sub-goals (G3, G4, G5) are present with linked Assumptions (A1, A2, A3), justifications (J1, J2, J3), and Solutions (E1, E2, E3). | Passed |

| Test ID | Test Objective | Description | Expected Output | actual XML output | Passed / Failed |
|---|---|---|---|---|---|
| T4 | Hazard Identification and Mitigation | Verify that the XML include hazards-related goals and corresponding mitigation for identified hazards in the U-SHIFT Control System. | The XML should include goals addressing specific hazards (Data processing failure, Command Interface loss, fault in diagnostic, sensor data misinterpretation, Loss of communication with subsystems). Each hazard should have a clear mitigation strategy and solutions to demonstrate its handling. | Sub-goals (G5, G6, G7, G8, G9) are present, each addressing specific hazards with linked contexts (C6-C10), assumptions (A4-A8), justifications (J4-J8), and solutions (E4-E8). | Passed |
| T5 | Regulatory Compliance Mapping | Ensure that the XML includes contexts, assumptions or justifications referencing compliance with regulatory standards | We expect the following standards: ISO 26262, UNECE R79 and SAE J3016 provided in the safety analysis and system design | No Node refers to the required safety standards. | Failed |

## 5.4 CASE 3: U-SHIFT COMMUNICATION SYSTEM

### 5.4.1 *Safety analysis and system description of the U-SHIFT Communication System*

The safety analysis and system description for the U-SHIFT Communication System are detailed in appendix A.7

### 5.4.2 *Generated safety cases of the U-Shift Communication System*

The U-SHIFT Communication System's safety analysis and system design was submitted to the trained model and the SCs generated by our trained model are shown in appendix A.8.

### 5.4.3 *Analysis of the U-SHIFT Communication System's safety cases in XML*

Table 5.3: Analysis of the U-SHIFT Communication System's safety cases in XML

| Test ID | Test Objective | Description | Expected Output | actual XML output | Passed / Failed |
|---|---|---|---|---|---|
| T1 | Top-goal Verification | Ensure that the top goal defines the overall Safety of the Communication System. | We expect to get something whose meaning is close to: "U-SHIFT Control system is acceptably safe", with child contexts, assumptions, justifications, and strategies | We got: Goal G1 with linked contexts (C1, C2), Assumption A1, Justification J1 and strategy S1. | Passed |
| T2 | Context Verification | Verify the inclusion of necessary contexts describing the system overview, boundaries and operational scenarios. | We expect information about system boundaries (definition), the subsystems and components, in accordance with the provided safety analysis and system design. | The context nodes C1 and C2 are included and describe the system overview and boundaries as specified. | Passed |
| T3 | Subsystem Decomposition | Validate that XML includes goals for Operator Interface, Passenger Notification, Data Encryption, and Monitoring subsystems with assumptions, justifications, and solutions. | We expect to get something whose meaning is close to: "Operator Interface subsystem is acceptably safe"; "¨Passenger Notification Subsystem is acceptably safe"; "Data Encryption Subsystem is acceptably safe" and "Monitoring Subsystem is acceptably safe", with child contexts, assumptions, justifications, and strategies | Sub-goals (G2, G4, G6) are present with linked Assumptions (A2, A4, A6), justifications (J2, J4, J6), and Contexts (C3, C5, C7). | Passed |

| Test ID | Test Objective | Description | Expected Output | actual XML output | Passed / Failed |
|---|---|---|---|---|---|
| T4 | Hazard Identification and Mitigation | Verify that the XML includes hazards-related goals and corresponding mitigations for identified hazards in the Communication System. | The XML should include goals addressing specific hazards (Network connectivity loss, data transmission delay, Unauthorized access, Data corruption, Incomplete passenger notifications). Each hazard should have a clear mitigation strategy and solutions to demonstrate its handling. | Sub-goals (G3, G5, G7) are present, each addressing specific hazards with linked contexts (C4, C6, C8), assumptions (A3, A5, A7), justifications (J3, J5, J7), and solutions (E1, E2, E3). | Passed |
| T5 | Regulatory Compliance Mapping | Ensure that the XML includes contexts, assumptions or justifications referencing compliance with regulatory standards | We expect the following standards: ISO 26262, UNECE R10, and SAE J3061 provided in the safety analysis and system design | No explicit node references standards directly in the provided safety cases. | Failed |

## 5.5 Conclusion of the testing phase

The testing phase demonstrated that the framework we created is able to generate GSN-based SCs in XML for the U-SHIFT Vehicle subsystems. Each test takes in consideration 5 important aspects (Top-Goal verification, Context Verification, Subsystem Decomposition, Hazard Identification and Mitigation, and Regulatory Compliance Mapping) in order to validate the generated SCs by the model. While the system successfully captured critical safety objectives, subsystem details, and hazard mitigation, the absence of explicit references to regulatory standards highlighted areas for improvement. Overall, the results confirm the system's capability to automate the generation of SCs and GSN Diagrams aligned with provided safety analyses and system descriptions.

## 5.6 Possible future improvements

A significant future improvement would be to enable the LLM itself to automatically decompose complex systems into smaller subsystems. And for each of the sub-system it should also generate itself structured, large-scale SCs in XML format. For a comprehensive system the USHIFT Vehicle, the model could be enhanced to autonomously identify subsystems (USHIFT Onboarding system, USHIFT Control System, USHIFT Communication system, USHIFT Emergency system, etc.) from initially provided safety analysis and system description, generating modular SCs as interconnected modular extensions. Each module would correspond to a subsystem and include its detailed SCs, allowing users to visualize the entire system in a hierarchy GSN Diagram. Clicking on a extension module node in the diagram would reveal its internal structure; facilitating scalability and improved usability for large systems. This improvement would require to extend the current GSN schema in XML. That means we will insert new GSN elements for extension modules and defining their interrelations with other existing GSN elements. This will lead to the regeneration of a new training dataset with domain specific SCs that will take in consideration the new define extensions elements of the GSN representation. This improvement will considerably extend the size of the generated SCs.

However, a key challenge is the ability of LLMs to generate such large-scale content due to token limitations. While models like Llama 3 (8B) exhibit strong generation capabilities, generating comprehensive SCs for entire system, including all subsystems, may exceed token constraints. Future enhancements should be geared to create mechanisms to iteratively produce content of each sub-system and later integrate all of them into forming an integrated XML structure.

More importantly, the following enhancements need to be considered further in order to increase the impact and applicability of the system:

**Automated Compliance Validation:** Integrate mechanisms automatically to validate generated SCs against regulatory standards; for example, ISO 26262, SOTIF, etc., so immediate assurance is provided for the generated specification..

**Domain-Specific Pre-training:** Fine-tuning the LLM with a larger and more diverse dataset of SCs across various industries (e.g aerospace, automotive, healthcare, etc. could enhance its contextual understanding and generation quality.

**Handling Incomplete or Ambiguous Inputs:** Improving the LLM's ability to infer and resolve ambiguities from incomplete safety analyses would make the tool more practical in real-world scenarios where data availability is limited.

**Multi-Modal Input Support:** This will allow the system to accept different input formats, like graphical models, tabular data and diagrams - among them, SysML and UML. These improvements are also expected to increase both the scalability and user-friendliness of the system while simultaneously increasing the scope of applicability of LLMs for large safety engineering contexts.

These improvements would not only make the system more scalable and usable but also push the state of the art in the applicability of LLMs in large-scale safety engineering. They are an important step toward the automation of hierarchy, modular SCs generation, which conforms to industry standards and capture the complexities of modern engineering systems.

# 6 CONCLUSION

This master's thesis presents a novel framework for the use of LLMs for generation of GSN-based SCs in XML format. The overall objective was to enable the automation of SCs and GSN diagrams generation from system descriptions and safety analyses. This work was supposed to improve the efficiency and accuracy of safety engineering processes by providing safety engineers with a starting point in their projects. To achieve this, we first created an XML syntax capable of representing GSN elements (Goals, Strategies, Assumptions, Justifications, Solutions) and SCs in a structured textual format. This developed XML-syntax was then utilized to craft system-specific prompts for the Llama-3-405B model, enabling the generation of a training dataset for fine-tuning the Llama-3-8B model. The generated dataset consists of three columns: system name, safety analysis and system description, and SCs in XML, comprising a total of 1000 rows. It has been published on the Hugging Face Hub and is publicly accessible. The pre-trained model was loaded from the repository of the company Unsloth on the Hugging Face Hub. Subsequently, the fine-tuning was performed using the powerful Unsloth framework, which enables training certain LLMs upto 5x faster with 70% less memory. The fine-tuned model was trained to interpret complex safety analyses and produce structured GSN-based SCs in XML-format, which were subsequently visualized as GSN diagrams using a custom-developed application capable of converting SCs in XML into graphical representations.

Once the model has been trained, its performance was evaluated using a real-world example: the U-SHIFT autonomous Vehicle developed by the German Aerospace Center (DLR). The complex architecture of the vehicle was manually decomposed into five subsystems based on the safety analysis and system description provided in the project documentation by the supervisor. Of these five subsystems, three were selected for testing (U-SHIFT Onboarding System, U-SHIFT control System, U-SHIFT Communication System). The testing phase aimed to assess the model's ability to generate GSN-based SCs in XML format, based on the supplied input (safety analysis and the system description). The testing phase demonstrated the effectiveness of the proposed system, with key evaluations validating the model's capability to generate safety cases accurately reflecting high-level goals, subsystem decompositions, risk mitigation's, and regulatory compliance requirements. The trained model successfully produced SCs for each subsystem of the U-SHIFT vehicle, with appropriate linkages between goals, contexts, assumptions, justifications, solutions, and strategies. However, elements such as explicit references to the regulatory compliance were noted as the areas of improvement in the future.

While the results are promising, the system faces challenges concerning the scalability due to token limits inherent in LLMs when dealing with extensive and complex systems like the U-SHIFT vehicle. Managing very large systems with multiple subsystems and inter-dependencies needs LLM architectures and iterative generation techniques advancements. Additionally, creating a syntax of GSN elements extensions (modular extensions) in XML is crucial for enabling hierarchical SCs generation and handle very large and complex systems. Another difficulty is the quality of the training data. As no relevant dataset on SCs, safety analyses and system design exist at this time, the dataset used to train Llama 3 8B was generated by Llama 3 405B. To obtain better performing model, it would be interesting to have a training dataset containing data from real-world systems and scenarios (not generated by a model). Future enhancements could address these limitations, including: Automated compliance validation (incorporating mechanisms to validate SCs against regulatory standards such as ISO 26262, and SOTIF, etc.), domain-specific pretraining (fine-tuning the model with diverse datasets from industries such as aerospace and healthcare to improve contextual understanding), handling incomplete inputs (enhancing the LLM's ability to interpret and resolve ambiguities from incomplete safety analyses) multi-model input support (enabling the system to process alternative input formats such as graphical models, and tabular data, to expand its usability across engineering domains), and the construction of a big training dataset based on real-world systems and scenarios (not generated by a bigger model).

However, because of the intrinsic tendency of LLMs to generate hallucinations (outputs that may be plausible but factually incorrect), the results from our model can never be fully trusted, even when trained on high-quality and domain-specific datasets. Consequently, it is imperative that the outputs generated by our model, undergo thorough review by qualified safety experts. These experts shall validate the completeness and correctness of the generated SCs, in order to ensure that they meet established safety standards and the specific requirements of the system under analysis. This review process is very important because it not only reduces the risks associated with potential inaccuracies but also reinforces the reliability and credibility of AI-based system in the safety engineering.

In conclusion, this research opens the way toward integration of LLMs in SCs generation and offers many valuable insights about the potential that AI-driven tools can be used for the field of safety engineering. Although challenges are still ahead, the results show great promise in terms of automating and developing SCs. This open wide avenues for both future researches and practical applications in safety-critical domains.

# BIBLIOGRAPHY


[1] N. A. Akbar, I. Darmayanti, S. M. Fati, and A. Muneer. "Deep Learning of a Pre-trained Language Model's Joke Classifier Using GPT-2". In: *Journal of Hunan University Natural Sciences* 48.8 (2021).

[2] Amazon. *What is Large Langue Model ?* Accessed: 2024-12-03. URL: https://aws.amazon.com/what-is/large-language-model/.

[3] A. S. Ayub. *XML, YAML, JSON; Differences and Similarities*. Accessed: 2024-11-19. 2024-05-14. URL: https://dev.to/anwar_sadat/ml-yaml-json-differences-and-similarities-3n17.

[4] P. Azunre. *Transfer learning for natural language processing*. Simon and Schuster, 2021.

[5] A. Balaguer, V. Benara, R. L. de Freitas Cunha, R. d. M. Estevão Filho, T. Hendry, D. Holstein, J. Marsman, N. Mecklenburg, S. Malvar, L. O. Nunes, et al. "RAG vs fine-tuning: Pipelines, tradeoffs, and a case study on agriculture". In: *arXiv e-prints* (2024), arXiv–2401.

[6] E. M. Bender, T. Gebru, A. McMillan-Major, and S. Shmitchell. "On the dangers of stochastic parrots: Can language models be too big?" In: *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency*. 2021, pp. 610–623.

[7] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al. "Language models are few-shot learners". In: *Advances in neural information processing systems* 33 (2020), pp. 1877–1901.

[8] J. Cámara, J. Troya, L. Burgueño, and A. Vallecillo. "On the assessment of generative AI in modeling tasks: an experience report with ChatGPT and UML". In: *Software and Systems Modeling* 22.3 (2023), pp. 781–793.

[9] M. B. Chaaben, L. Burgueño, and H. Sahraoui. "Towards using few-shot prompt learning for automating model completion". In: *2023 IEEE/ACM 45th International Conference on Software Engineering: New Ideas and Emerging Results (ICSE-NIER)*. IEEE. 2023, pp. 7–12.

[10] S. Charalampidou, A. Zeleskidis, and I. M. Dokas. "Hazard analysis in the era of AI: Assessing the usefulness of ChatGPT4 in STPA hazard analysis". In: *Safety Science* 178 (2024), p. 106608.

[11] A. Chavan, R. Magazine, S. Kushwaha, M. Debbah, and D. Gupta. "Faster and Lighter LLMs: A Survey on Current Challenges and Way Forward". In: *arXiv preprint arXiv:2402.01799* (2024).

[12] M. Chelouati, A. Boussif, J. Beugin, and E.-M. El Koursi. "Graphical safety assurance case using Goal Structuring Notation (GSN)—challenges, opportunities and a framework for autonomous trains". In: *Reliability Engineering & System Safety* 230 (2023), p. 108933.

[13] B. Chen, Z. Zhang, N. Langrené, and S. Zhu. "Unleashing the potential of prompt engineering in Large Language Models: a comprehensive review". In: *arXiv preprint arXiv:2310.14735* (2023).

[14] B. Chen, K. Chen, S. Hassani, Y. Yang, D. Amyot, L. Lessard, G. Mussbacher, M. Sabetzadeh, and D. Varró. "On the use of GPT-4 for creating goal models: an exploratory study". In: *2023 IEEE 31st International Requirements Engineering Conference Workshops (REW)*. IEEE. 2023, pp. 262–271.

[15] B. Cronin. "Annual review of information science and technology". In: (2004).

[16] F. Cuconasu, G. Trappolini, F. Siciliano, S. Filice, C. Campagnano, Y. Maarek, N. Tonellotto, and F. Silvestri. "The power of noise: Redefining retrieval for rag systems". In: *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2024, pp. 719–729.

[17] E. Denney and G. Pai. "Evidence arguments for using formal methods in software certification". In: *2013 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW)*. IEEE. 2013, pp. 375–380.

[18] E. Denney and G. Pai. "Automating the assembly of aviation safety cases". In: *IEEE Transactions on Reliability* 63.4 (2014), pp. 830–849.

[19] E. Denney and G. Pai. "Tool support for assurance case development". In: *Automated Software Engineering* 25.3 (2018), pp. 435–499.

[20] T. Dettmers, M. Lewis, Y. Belkada, and L. Zettlemoyer. "Gpt3. int8 (): 8-bit matrix multiplication for transformers at scale". In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 30318–30332.

[21] T. Dettmers, A. Pagnoni, A. Holtzman, and L. Zettlemoyer. "Qlora: Efficient finetuning of quantized llms". In: *Advances in Neural Information Processing Systems* 36 (2024).

[22] DhanushKumar. *Fine-tuning large language models (LLMs)*. Accessed: 2024-11-19. 2024-06-05. URL: https://medium.com/@danushidk507/fine-tuning-large-language-models-llms-1aa5461e20ac.

[23] S. Diemert and J. H. Weber. "Can large language models assist in hazard analysis?" In: *International Conference on Computer Safety, Reliability, and Security*. Springer. 2023, pp. 410–422.

[24] A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Yang, A. Fan, et al. "The llama 3 herd of models". In: *arXiv preprint arXiv:2407.21783* (2024).

[25] L. Floridi and M. Chiriatti. "GPT-3: Its nature, scope, limits, and consequences". In: *Minds and Machines* 30 (2020), pp. 681–694.

[26] S. Frieder, L. Pinchetti, R.-R. Griffiths, T. Salvatori, T. Lukasiewicz, P. Petersen, and J. Berner. "Mathematical capabilities of chatgpt". In: *Advances in neural information processing systems* 36 (2024).

[27] I. O. Gallegos, R. A. Rossi, J. Barrow, M. M. Tanjim, S. Kim, F. Dernoncourt, T. Yu, R. Zhang, and N. K. Ahmed. "Bias and fairness in large language models: A survey". In: *Computational Linguistics* (2024), pp. 1–79.

[28] T. A. C. W. Group. *Goal Structuring Notation; Community Standard; Version 3; SCSC-141C*. Accessed: 2024-11-18. n.d. URL: https://scsc.uk/scsc-141c.

[29] M. U. Hadi, Q. Al Tashi, A. Shah, R. Qureshi, A. Muneer, M. Irfan, A. Zafar, M. B. Shaikh, N. Akhtar, J. Wu, et al. "Large language models: a comprehensive survey of its applications, challenges, limitations, and future prospects". In: *Authorea Preprints* (2024).

[30] C. Hartsell, N. Mahadevan, A. Dubey, and G. Karsai. "Automated method for assurance case construction from system design models". In: *2021 5th International Conference on System Reliability and Safety (ICSRS)*. IEEE. 2021, pp. 230–239.

[31] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen. "Lora: Low-rank adaptation of large language models". In: *arXiv preprint arXiv:2106.09685* (2021).

[32] X. Huang, W. Ruan, W. Huang, G. Jin, Y. Dong, C. Wu, S. Bensalem, R. Mu, Y. Qi, X. Zhao, et al. "A survey of safety and trustworthiness of large language models through the lens of verification and validation". In: *Artificial Intelligence Review* 57.7 (2024), p. 175.

[33] *Hugging Face – Wikipedia*. https://de.wikipedia.org/wiki/Hugging_Face. Accessed: 2024-08-29.

[34] C. Jeong. "A study on the implementation of generative ai services using an enterprise data-based llm application architecture". In: *arXiv preprint arXiv:2309.01105* (2023).

[35] B.-H. Juang and L. R. Rabiner. "Automatic speech recognition–a brief history of the technology development". In: *Georgia Institute of Technology. Atlanta Rutgers University and the University of California. Santa Barbara* 1.67 (2005), p. 1.

[36] T. Kelly and R. Weaver. "The goal structuring notation–a safety argument notation". In: *Proceedings of the dependable systems and networks 2004 workshop on assurance cases*. Vol. 6. Citeseer Princeton, NJ. 2004.

[37] T. P. Kelly et al. "Arguing safety: a systematic approach to managing safety cases". PhD thesis. Citeseer, 1999.

[38] A. Kovačević and D. Kečo. "Bidirectional LSTM networks for abstractive text summarization". In: *Advanced Technologies, Systems, and Applications VI: Proceedings of the International Symposium on Innovative and Interdisciplinary Applications of Advanced Technologies (IAT) 2021*. Springer. 2022, pp. 281–293.

[39] E. D. Liddy. "Natural language processing". In: (2001).

[40] I. Loshchilov, F. Hutter, et al. "Fixing weight decay regularization in adam". In: *arXiv preprint arXiv:1711.05101* 5 (2017).

[41] M. Macias. "Finetuning and Improving Prediction Results of LLMs Using Synthetic Data". In: (2024).

[42] M. Maksimov, S. Kokaly, and M. Chechik. "A survey of tool-supported assurance case assessment techniques". In: *ACM Computing Surveys (CSUR)* 52.5 (2019), pp. 1–34.

[43] B. Mann, N Ryder, M Subbiah, J Kaplan, P Dhariwal, A Neelakantan, P Shyam, G Sastry, A Askell, S Agarwal, et al. "Language models are few-shot learners". In: *arXiv preprint arXiv:2005.14165* 1 (2020).

[44] G. Marvin, N. Hellen, D. Jjingo, and J. Nakatumba-Nabende. "Prompt engineering in large language models". In: *International conference on data intelligence and cognitive informatics*. Springer. 2023, pp. 387–402.

[45] J Mathav Raj, V. Kushala, and Y. G. Harikrishna Warrier. "Fine Tuning LLM for Enterprise: Practical Guidelines and Recommendations". In: *arXiv preprint arXiv:2404.10779* (2024).

[46] NvidiaNIM. *API source code*. Accessed: 2024-11-20. URL: https://www.nvidia.com/en-us/ai/.

[47] J. Oppenlaender. "A taxonomy of prompt modifiers for text-to-image generation". In: *Behaviour & Information Technology* (2023), pp. 1–14.

[48] R. Patil and V. Gudivada. "A review of current trends, techniques, and challenges in large language models (llms)". In: *Applied Sciences* 14.5 (2024), p. 2074.

[49] Y. Qi, X. Zhao, S. Khastgir, and X. Huang. "safety analysis in the era of large language models: a case study of STPA using ChatGPT". In: *arXiv preprint arXiv:2304.01246* (2023).

[50] A. Radford. "Improving language understanding by generative pre-training". In: (2018).

[51] S. Ramakrishna, C. Hartsell, A. Dubey, P. Pal, and G. Karsai. "A methodology for automating assurance case generation". In: *arXiv preprint arXiv:2003.05388* (2020).

[52] A. Robertson. "European Organisation for the Safety of Air Navigation (EUROCONTROL)". In: *Annuaire Europeen/European Yearbook: Vol. XXVII* (1981), pp. 618–629.

[53] A. R. Sajun, I. Zualkernan, and D. Sankalpa. "A Historical Survey of Advances in Transformer Architectures". In: *Applied Sciences* 14.10 (2024). ISSN: 2076-3417. DOI: 10.3390/app14104316. URL: https://www.mdpi.com/2076-3417/14/10/4316.

[54] N. Selviandro, R. Hawkins, and I. Habli. "A visual notation for the representation of assurance cases using sacm". In: *Model-Based Safety and Assessment: 7th International Symposium, IMBSA 2020, Lisbon, Portugal, September 14–16, 2020, Proceedings 7*. Springer. 2020, pp. 3–18.

[55] M. Sivakumar. "DESIGN AND AUTOMATIC GENERATION OF SAFETY CASES OF ML-ENABLED AUTONOMOUS DRIVING SYSTEMS". In: (2024).

[56] M. Sivakumar, A. B. Belle, J. Shan, and K. K. Shahandashti. "GPT-4 and Safety Case Generation: An Exploratory Analysis". In: *arXiv preprint arXiv:2312.05696* (2023).

[57] M. Smetana, L. Salles de Salles, I. Sukharev, and L. Khazanovich. "Highway Construction Safety Analysis Using Large Language Models". In: *Applied Sciences* 14.4 (2024), p. 1352.

[58] J. Spriggs. *GSN-the goal structuring notation: A structured approach to presenting arguments*. Springer Science & Business Media, 2012.

[59] E. Strubell, A. Ganesh, and A. McCallum. "Energy and policy considerations for modern deep learning research". In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 34. 09. 2020, pp. 13693–13696.

[60] Unsloth. *Google colab templates*. Accessed: 2024-11-20. URL: https://huggingface.co/unsloth.

[61] K. VM, H. Warrier, Y. Gupta, et al. "Fine Tuning LLM for Enterprise: Practical Guidelines and Recommendations". In: *arXiv preprint arXiv:2404.10779* (2024).

[62] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. "Attention is all you need". In: *Advances in neural information processing systems* 30 (2017).

[63] T. Viger, L. Murphy, S. Diemert, C. Menghi, A. Di, and M. Chechik. "Supporting Assurance Case Development Using Generative AI". In: *SAFECOMP 2023, Position Paper*. 2023.

[64] T. E. Wang, C. Oh, M. Low, I. Amundson, Z. Daw, A. Pinto, M. L. Chiodo, G. Wang, S. Hasan, R. Melville, et al. "Computer-aided generation of assurance cases". In: *International Conference on Computer Safety, Reliability, and Security*. Springer. 2023, pp. 135–148.

[65] Y. Wardat, M. A. Tashtoush, R. AlAli, and A. M. Jarrah. "ChatGPT: A revolutionary tool for teaching and learning mathematics". In: *Eurasia Journal of Mathematics, Science and Technology Education* 19.7 (2023), em2286.

[66] M. Weyssow, H. Sahraoui, and E. Syriani. "Recommending metamodel concepts during modeling activities with pre-trained language models". In: *Software and Systems Modeling* 21.3 (2022), pp. 1071–1089.

[67] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, et al. "Transformers: State-of-the-art natural language processing". In: *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*. 2020, pp. 38–45.

[68] S. Wu, O. Irsoy, S. Lu, V. Dabravolski, M. Dredze, S. Gehrmann, P. Kambadur, D. Rosenberg, and G. Mann. "Bloomberggpt: A large language model for finance". In: *arXiv preprint arXiv:2303.17564* (2023).

[69] Y. Wu. "Google's neural machine translation system: Bridging the gap between human and machine translation". In: *arXiv preprint arXiv:1609.08144* (2016).

[70] H. Yang, X.-Y. Liu, and C. D. Wang. "Fingpt: Open-source financial large language models". In: *arXiv preprint arXiv:2306.06031* (2023).

[71] elastic. *What are Large Langue Models (LLMs) ?* Accessed: 2024-12-03. URL: https://www.elastic.co/what-is/large-language-models.

# A APPENDIX

## A.1 System prompt for the generation of safety analyses and system description

Listing A.1: Prompt for the generation of safety analyses and system description

```
SYSTEM_PROMPT_Saf_Anal_Sys_Desc = """

You are a systems and security engineer with over 10 years’ experience in
    ↪ developing safety analyses for critical systems.
Your task is to GENERATE a high-level safety analysis and system description,
    ↪ which must be as complete as possible, i.e. cover all the various
    ↪ aspects required,
such as..: System description and boundaries, operational environment,
    ↪ functional and performance requirements, safety critical functions,
hazards analysis, System Architecture and design, Implementation details,
    ↪ verification and validation, Maintenance and operation,
Regulatory and compliance Information, Historical data and lessons learned,
    ↪ Stakeholders involved.


<--- Here is an example of system description and Safety Analysis --->

### System Description and Safety Analysis: Autonomous Emergency Braking
    ↪ System (AEBS)

---

#### 1. System Definition

**System Overview**:
The Autonomous Emergency Braking System (AEBS) is a driver-assist safety
    ↪ feature designed to automatically detect potential collisions and
    ↪ apply the v e h i c l e s brakes without driver input. AEBS primarily
    ↪ operates in urban environments, scanning the road for obstacles such
    ↪ as pedestrians, cyclists, and vehicles. Upon detection of an imminent
    ↪ collision, the system calculates the risk and intervenes by activating
    ↪ the braking system to either mitigate or avoid the impact. The AEBS is
    ↪ part of the v e h i c l e s broader safety architecture and integrates
    ↪ with other onboard systems like adaptive cruise control and
    ↪ lane-keeping assistance.

**System Boundaries**:
```

```
- **Subsystems**:
  - **Sensing Subsystem**: A combination of radar, camera, and LiDAR sensors that work together to detect obstacles in the vehicles path.
  - **Decision-Making Subsystem**: Real-time processing unit (CPU/GPU) that evaluates sensor data to assess collision risk.
  - **Actuation Subsystem**: The braking system, which includes an electro-mechanical braking actuator, hydraulic braking support, and brake fluid monitoring systems.
  - **User Interaction Subsystem**: Interfaces such as dashboard alerts, audio warnings, and manual brake overrides.

- **Components**:
  - **Sensors**: The AEBS uses a **front-facing radar** with a detection range of 80 meters, a **wide-angle camera** for detecting pedestrians and smaller objects, and **LiDAR** to provide depth and object classification.
  - **Processing Unit**: A high-speed processing unit analyzes sensor data and calculates the time-to-collision (TTC), then determines whether to initiate braking.
  - **Braking Actuators**: Electro-mechanical actuators are responsible for applying the brakes based on signals from the processing unit. Redundancy is built into the braking system to ensure functionality in case of failure.

---

#### 2. Operational Environment

**Physical Environment**:
The AEBS is optimized for urban and suburban environments where pedestrians, cyclists, and vehicles frequently interact. It must function under various physical conditions:
- **Road Types**: Asphalt, concrete, and cobblestone surfaces.
- **Weather Conditions**: The system operates effectively in clear, rainy, foggy, snowy, and icy weather. Radar sensors can detect obstacles through fog or rain, but the system may face challenges with occlusion caused by heavy snowfall.
- **Lighting Conditions**: AEBS must work during the day, night, and twilight, adjusting to varying light levels with the help of infrared capabilities.

**Operational Scenarios**:
- **Scenario 1**: A pedestrian unexpectedly crosses the street in front of the vehicle while the car is traveling at 40 km/h in an urban area.
- **Scenario 2**: A cyclist moves into the vehicles lane from a sidewalk, triggering a potential collision.
- **Scenario 3**: The car ahead of the vehicle brakes abruptly, requiring an emergency braking response to avoid a rear-end collision.
- **Scenario 4**: A child running from behind a parked car into the street, requiring an immediate braking response.
```

```

**Human Interaction**:
The system warns the driver before automatic braking through:
- **Auditory Alerts**: Warning chimes escalate in volume as a collision becomes more imminent.
- **Visual Alerts**: Dashboard warning lights and hazard symbols flash, indicating a potential need for driver intervention.
- **Manual Override**: The driver can override the AEBS by manually applying the brakes or steering the vehicle away from the obstacle.

---

#### 3. Functional and Performance Requirements

**Functional Requirements**:
- **Obstacle Detection**: AEBS must continuously monitor the road ahead for pedestrians, cyclists, vehicles, and static obstacles like walls and barriers.
- **Collision Risk Assessment**: The processing unit must compute the probability of a collision based on the speed, distance, and trajectory of detected objects.
- **Braking Activation**: When a high risk of collision is determined, AEBS should engage the braking system within 200 milliseconds.
- **Driver Alerts**: AEBS should provide a warning to the driver at least 1.5 seconds before automatically applying the brakes unless an immediate intervention is necessary.

**Performance Requirements**:
- **Detection Range**: Radar and LiDAR must detect objects up to 80 meters ahead, with accurate object classification within a 50-meter range.
- **Braking Precision**: AEBS must achieve a stopping distance reduction of at least 30% in emergency situations.
- **Latency**: Processing latency for object detection and decision-making must not exceed 100 milliseconds, and total response time (from detection to full braking) must be under 250 milliseconds.
- **Weather Performance**: AEBS must maintain at least 90% detection accuracy in adverse weather conditions, including heavy rain and moderate snowfall.

---

#### 4. Safety-Critical Functions

**Identification**:
- **Function 1**: Obstacle detection and classification combining data from radar, LiDAR, and camera sensors to identify pedestrians, cyclists, and vehicles.
- **Function 2**: Collision risk assessment and decision-making using time-to-collision (TTC) calculations and trajectory prediction algorithms.
```

```
- **Function 3**: Emergency braking activating the braking actuator system based on collision risk assessment.
- **Function 4**: Driver warning issuing timely alerts for manual intervention when a collision risk is identified.

**Failure Modes**:
- **Mode 1**: **Sensor Blindness** failure to detect objects due to sensor malfunction or occlusion (e.g., dirt on the camera lens or radar blockage in heavy rain).
- **Mode 2**: **False Detection** system mistakenly detects an obstacle when none exists, triggering unnecessary braking.
- **Mode 3**: **Braking System Failure** failure of the braking actuator or hydraulic system, preventing the AEBS from engaging the brakes.
- **Mode 4**: **Processing Delay** system latency causing a delay in braking, increasing the likelihood of a collision.

---

#### 5. System Architecture and Design

**Architecture Overview**:
The system is designed with modular components that communicate over a high-speed vehicle network. It consists of:
- **Sensor Layer**: Multiple sensor modalities (radar, camera, LiDAR) feeding data to the processing unit.
- **Processing Layer**: A real-time processor running a collision avoidance algorithm that computes TTC and trajectory prediction.
- **Actuation Layer**: The braking system, including actuators and hydraulic support, responsible for executing the braking decision.

**Design Principles**:
- **Redundancy**: Critical components, such as braking actuators and sensor arrays, have built-in redundancy to ensure that a failure in one subsystem does not compromise the safety of the overall system.
- **Fail-safe Operation**: In the event of a system fault, AEBS defaults to alerting the driver and transferring control back to manual driving.

**Safety Mechanisms**:
- **Cross-sensor Validation**: Sensor data is cross-validated (e.g., radar and camera) to reduce false positives.
- **Self-diagnostics**: Continuous monitoring of sensor health, communication links, and braking system integrity.

---

#### 6. Implementation Details

**Software**:
The AEBS software suite is divided into three core modules:
```

```
1. **Sensor Fusion Module**: Aggregates data from radar, camera, and LiDAR to provide a comprehensive view of the road ahead.
2. **Collision Prediction Algorithm**: Predicts potential collisions based on trajectory analysis and speed profiles of detected objects.
3. **Braking Control Software**: Interacts with the vehicles braking system to apply appropriate braking force based on the calculated risk.

**Hardware**:
- **Sensing Hardware**: Radar (80m range), LiDAR (3D mapping within 50m), and a high-definition camera (wide-angle for peripheral vision).
- **Processing Unit**: A high-speed real-time processor (e.g., NVIDIA Drive AGX) capable of processing sensor data and executing the collision prediction algorithm within 100ms.
- **Braking Actuator**: An electro-mechanical actuator with a hydraulic backup, ensuring quick brake engagement and redundancy in case of failure.

**Integration and Configuration Management**:
AEBS is integrated with the vehicle’s central control system via the Controller Area Network (CAN) bus. Configuration management includes regular software updates, version control, and secure boot processes to ensure system integrity.

---

#### 7. Verification and Validation

**Testing Procedures**:
- **Unit Testing**: Each individual component (sensor, processing unit, braking actuator) is tested for functionality under controlled lab conditions.
- **Integration Testing**: Full system testing in real-world scenarios, including vehicle operation in urban environments with pedestrian crossings and sudden stops.
- **Stress Testing**: Evaluate the system under extreme conditions, such as dense fog, torrential rain, and low-visibility night driving.

**Validation Methods**:
- **Simulation Testing**: Extensive use of traffic simulations to test the systems response to a variety of urban driving scenarios, including high-density traffic and unpredictable pedestrian behavior.
- **Field Testing**: Real-world testing in city environments, with particular focus on high-traffic areas where sudden braking is often required.

**Test Results**:
- Pedestrian detection accuracy: 98%.
- Braking response time: Achieved within 180ms in 90% of test cases.
- False positive rate: Less than 5% in controlled tests.

---
```

```
#### 8. Maintenance and Operation

**Maintenance Procedures**:
- **Sensor Calibration**: Sensors require regular calibration to ensure
    ↪ proper alignment and accurate obstacle detection.
- **Braking System Inspection**: Periodic checks on hydraulic lines, brake
    ↪ fluid levels, and actuator function.
- **Software Updates**: Regular software patches to address bugs, improve
    ↪ performance, and introduce new features based on real-world feedback.

**Operational Procedures**:
- AEBS is automatically activated when the vehicle is in motion within urban
    ↪ environments but can be manually disengaged if desired by the driver.
- The system logs all interventions and warnings for later review by
    ↪ maintenance personnel or vehicle operators.

**Incident Response**:
- In the event of an AEBS intervention, a report is generated for diagnostic
    ↪ review, detailing the event, system actions, and sensor readings at
    ↪ the time of the incident.

---

#### 9. Regulatory and Compliance Information

**Applicable Standards**:
- **ISO 26262**: Functional safety standard for road vehicles.
- **UNECE Regulation 152**: Governing the requirements for advanced emergency
    ↪ braking systems (AEBS) in passenger and commercial vehicles.
- **SAE J3016**: Classification of driving automation levels (level 0-5),
    ↪ with AEBS considered as a Level 1/2 automation feature.

**Compliance Strategy**:
- The AEBS is developed in compliance with automotive safety standards to
    ↪ ensure it meets or exceeds regulatory requirements. Independent audits
    ↪ and safety assessments are conducted to verify compliance.

---

#### 10. Safety Analysis

**Purpose**:
The purpose of this safety analysis is to identify potential hazards, assess
    ↪ the associated risks, and propose mitigations to ensure that the AEBS
    ↪ meets safety requirements.

---

#### 10.1 Hazard Identification
```

```

| **ID**   | **Hazard**                          | **Description**                                     |
|---------|-----------------------------------|-----------------------------------------------------------------|
| **H1**   | Pedestrian not detected            | AEBS fails to detect a pedestrian in time to avoid collision.   |
| **H2**   | False positive braking             | AEBS activates braking without an actual hazard.                |
| **H3**   | Braking system failure             | Braking actuators or hydraulic system fail to engage brakes.    |
| **H4**   | Sensor malfunction in bad weather  | Radar or camera fails in adverse weather conditions.            |
| **H5**   | Delayed braking response           | System response is delayed, increasing the likelihood of impact. |

---

#### 10.2 Risk Assessment

| **Hazard ID** | **Likelihood** | **Severity** | **Risk Rating** | **Mitigation Strategy**                                 |
|--------------|---------------|-------------|----------------|----------------------------------------------------------|
| **H1**        | Likely (3)     | Severe (4)   | High (12)       | Cross-validation using multiple sensor types (radar + LiDAR). |
| **H2**        | Occasional (2)  | Moderate (3) | Medium (6)      | Implement AI-based object recognition to reduce false alarms. |
| **H3**        | Rare (1)        | Severe (4)   | Medium (4)      | Regular brake system inspections and redundancy mechanisms.   |
| **H4**        | Occasional (2)  | Moderate (3) | Medium (6)      | Sensor health checks and self-calibration routines.           |
| **H5**        | Unlikely (1)    | Critical (5) | High (5)        | Optimize processing speed, use high-performance processors.   |

---

#### 10.3 Risk Mitigation Strategies

- **H1 - Pedestrian Not Detected**: Enhance pedestrian detection by using a combination of infrared sensors and advanced machine learning algorithms.
- **H2 - False Positive Braking**: Improve object classification and trajectory prediction to avoid unnecessary braking.
- **H3 - Braking System Failure**: Introduce failover systems that engage secondary braking mechanisms if the primary system fails.
- **H4 - Sensor Malfunction in Bad Weather**: Use sensor cleaning mechanisms and advanced algorithms to handle degraded sensor input in inclement weather.



```

```
<--- Here is another example of system description and Safety Analysis --->


### System Description and Safety Analysis: Aircraft Collision Avoidance System (ACAS)

---

#### 1. System Definition

**System Overview**:
The Aircraft Collision Avoidance System (ACAS) is an onboard safety system designed to prevent mid-air collisions by issuing alerts and executing avoidance maneuvers. It monitors surrounding airspace to detect aircraft and calculates potential collision risks. If necessary, ACAS advises the pilot or automatically takes control to execute evasive maneuvers.

**System Boundaries**:
- **Subsystems**:
  - **Surveillance Subsystem**: Radar, ADS-B, and transponder systems for aircraft detection.
  - **Decision-Making Subsystem**: Real-time processor for risk assessment and avoidance maneuver planning.
  - **Actuation Subsystem**: Flight control interfaces for maneuver execution.
  - **Pilot Interface**: Alerts, guidance, and manual override options for pilots.

- **Components**:
  - **Radar and ADS-B**: Sensors to detect nearby aircraft.
  - **Processing Unit**: Collision prediction and avoidance decision-making.
  - **Flight Control System**: Executes avoidance maneuvers.
  - **Cockpit Interface**: Alerts the pilot with visual and audio cues.

---

#### 2. Operational Environment

**Physical Environment**:
ACAS operates in all phases of flight and in diverse airspace environments, including low and high altitudes, varying weather conditions, and both day/night scenarios. The system must handle situations such as dense air traffic, weather disruptions, and varying terrain.

**Operational Scenarios**:
- **Scenario 1**: Two aircraft on converging flight paths.
- **Scenario 2**: A fast-moving aircraft approaching from behind.
- **Scenario 3**: Near-miss during takeoff or landing in foggy conditions.
- **Scenario 4**: Sudden maneuver from an adjacent aircraft at high altitudes.
```

```
---

#### 3. Functional and Performance Requirements

**Functional Requirements**:
- **Continuous Detection**: Detect aircraft within a 50 km range.
- **Collision Prediction**: Assess risk based on trajectories, speed, and altitude.
- **Maneuver Execution**: Provide real-time avoidance strategies within 5 seconds.
- **Alerting**: Issue timely warnings to the pilot with visual and audio cues.

**Performance Requirements**:
- **Detection Accuracy**: 99% detection reliability under all weather conditions.
- **Maneuver Execution Time**: Within 3 seconds of collision detection.
- **Separation Distance**: Maintain a minimum of 300 meters vertical or 1,000 meters horizontal clearance during evasive maneuvers.

---

#### 4. Safety-Critical Functions

- **Collision Detection and Avoidance**: Real-time prediction of collision risks and automatic execution of maneuvers.
- **Pilot Alerts**: Warning system to inform pilots of high-risk situations and recommend actions.
- **Manual Override**: Ensures the pilot can take over at any time.

---

#### 5. System Architecture and Design

ACAS is designed with redundancy in both sensors and control systems. It uses a layered architecture with real-time data fusion from multiple sensors, decision-making logic, and control outputs to the flight systems.

- **Surveillance Layer**: Gathers real-time data from radar and ADS-B.
- **Processing Layer**: Computes collision risks.
- **Control Layer**: Executes avoidance maneuvers.
- **Human Interface Layer**: Communicates alerts and recommendations to pilots.

---

#### 6. Implementation Details

- **Software**: Real-time decision algorithms for collision avoidance and trajectory analysis.
```

```
- **Hardware**: Radar, ADS-B receivers, transponders, and flight control actuators.

---

#### 7. Verification and Validation

**Testing Methods**:
- **Simulation Testing**: Extensive use of simulations for various air traffic densities and collision scenarios.
- **Flight Testing**: Testing in controlled airspace for real-world validation.

---

#### 8. Maintenance and Operation

**Maintenance**: Regular calibration of sensors and software updates. Periodic flight control system inspections.

---

#### 9. Regulatory and Compliance Information

**Standards**:
- **RTCA DO-185B**: Minimum operational performance standards for ACAS.
- **FAA and ICAO Regulations**: Compliance with international safety standards.

---

#### 10. Safety Analysis

The purpose of this safety analysis is to identify and mitigate risks associated with the operation of the ACAS system.

---

#### 10.1 Hazard Identification

| **ID**  | **Hazard**                          | **Description**                                  |
|---------|-------------------------------------|--------------------------------------------------------------
| **H1**  | Aircraft not detected               | ACAS fails to detect a nearby aircraft in time to avoid collision.|
| **H2**  | False positive maneuver             | ACAS recommends an unnecessary avoidance maneuver.               |
| **H3**  | Incorrect trajectory calculation    | The system miscalculates collision risks due to faulty data input.|
```

```
| **H4**  | Sensor failure in storm           | Radar or ADS-B malfunctions during severe weather.          |
| **H5**  | Delayed response                  | ACAS takes too long to issue a maneuver recommendation.        |
| **H6**  | Pilot override malfunction        | Pilot unable to override automatic controls in a critical situation.|
| **H7**  | Inaccurate altitude data          | Altitude discrepancies cause incorrect avoidance maneuvers.      |
| **H8**  | Actuation system failure          | Control surfaces fail to execute the maneuver.                   |
| **H9**  | Communication failure             | Loss of communication between ACAS and other aircraft systems.   |
| **H10** | Overloading in high-density traffic| ACAS fails to manage multiple collision risks in busy airspace.  |
| **H11** | Cross-traffic misinterpretation   | ACAS misinterprets data from aircraft crossing perpendicular paths.|
| **H12** | Mid-maneuver collision risk       | ACAS avoidance maneuver leads to a new collision risk.           |
| **H13** | Faulty radar calibration          | Radar calibration errors lead to incorrect distance measurements.|
| **H14** | Multiple subsystem failures       | Simultaneous failure of two or more subsystems, causing loss of functionality.|
| **H15** | False alert during low traffic    | ACAS triggers an alert during low traffic or low-risk scenarios. |

---

#### 10.2 Risk Assessment

| **Hazard ID** | **Likelihood** | **Severity** | **Risk Rating** | **Mitigation Strategy**                          |
|---------------|----------------|--------------|-----------------|---------------------------------------------------
| **H1**        | Likely (3)     | Catastrophic (5) | High (15)       | Improve sensor accuracy and redundancy with cross-checking.   |
| **H2**        | Occasional (2)  | Moderate (3) | Medium (6)      | Refine algorithms to reduce false positives.               |
| **H3**        | Occasional (2)  | Severe (4)   | High (8)        | Cross-verify data from different sensors for consistency.     |
| **H4**        | Occasional (2)  | Critical (5) | Medium (10)     | Improve sensor designs to withstand extreme weather.          |
| **H5**        | Unlikely (1)    | Critical (5) | Medium (5)      | Reduce processing time through software optimizations.        |
| **H6**        | Rare (1)        | Severe (4)   | Medium (4)      | Ensure reliable manual override mechanisms.                   |
| **H7**        | Occasional (2)  | Critical (5) | Medium (10)     | Validate altitude data with cross-reference checks.          |
| **H8**        | Rare (1)        | Severe (4)   | Medium (4)      | Introduce redundant control actuation systems.               |
| **H9**        | Rare (1)        | Critical (5) | Medium (5)      | Enhance system communication protocols with error-checking.   |
```

```
| **H10**       | Likely (3)      | Catastrophic (5) | High (15)       | Implement traffic prioritization algorithms.                 |
| **H11**       | Occasional (2)  | Moderate (3) | Medium (6)       | Develop smarter data interpretation for cross-traffic.       |
| **H12**       | Rare (1)        | Critical (5) | Medium (5)       | Continuous risk monitoring during avoidance maneuvers.        |
| **H13**       | Occasional (2)  | Critical (5) | Medium (10)      | Schedule regular radar calibration checks.                   |
| **H14**       | Rare (1)        | Catastrophic (5) | Medium (5)       | Increase subsystem redundancy and independent failure detection.|
| **H15**       | Rare (1)        | Low (2)      | Low (2)          | Implement adaptive alert sensitivity based on traffic density. |

---

#### 10.3 Risk Mitigation Strategies

- **H1 - Aircraft Not Detected**: Improve sensor range and add complementary detection technologies such as infrared sensors.
- **H2 - False Positive Maneuver**: Enhance decision-making algorithms to differentiate better between real threats and false alarms.
- **H3 - Incorrect Trajectory Calculation**: Implement multiple data sources and introduce cross-verification mechanisms for trajectory analysis.
- **H4 - Sensor Failure in Storm**: Introduce weather-hardened sensors and adaptive algorithms to adjust for signal degradation.
- **H5 - Delayed Response**: Invest in faster processing units and optimize code for quick decision-making in high-pressure scenarios.
- **H6 - Pilot Override Malfunction**: Ensure manual override testing in various failure modes and provide alternative manual controls.
- **H7 - Inaccurate Altitude Data**: Cross-verify altitudes with secondary systems and sensors like GPS.
- **H8 - Actuation System Failure**: Add a backup control system with the ability to execute evasive maneuvers if the primary system fails.
- **H9 - Communication Failure**: Design a robust, redundant communication protocol with error correction.
- **H10 - Overloading in High-Density Traffic**: Improve system processing by using AI models to prioritize high-risk targets.
- **H11 - Cross-Traffic Misinterpretation**: Introduce trajectory prediction models that better handle perpendicular flight paths.
- **H12 - Mid-Maneuver Collision Risk**: Monitor ongoing situations post-maneuver and adjust dynamically to avoid new risks.
- **H13 - Faulty Radar Calibration**: Enforce frequent radar system calibrations and introduce automated calibration verification systems.
- **H14 - Multiple Subsystem Failures**: Ensure robust safety nets that handle simultaneous failures without compromising safety.
- **H15 - False Alert During Low Traffic**: Calibrate the system to adjust alert sensitivity based on traffic density and conditions.


<--- output requirements --->
```

```
- The generated safety analysis should have minimum 10 identified Hazards
- Provide as much information as possible, to enable the generation of GSN elements that are rich in form and content (justification, assumption, context, goals, solutions), even on the lowest level GSN elements in the diagram.
```

## A.2 SYSTEM PROMPT FOR THE GENERATION OF SAFETY CASES IN XML BASED ON GSN

Listing A.2: Prompt for the generation of safety cases in XML based on GSN

```

SYSTEM_PROMPT_SAFETY_CASES = """ You are now a Senior System Safety Engineer with over 15 years of experience in designing safety cases for safety-critical systems in GSN (Goal Structuring Notation) and using XML representation. Your task is to generate safety cases in XML format using Goal Structuring Notation (GSN) framework.

**Here is a kind reminder about Safety Cases and GSN Elements:**
A safety case is a structured argument, supported by evidence, that a system is acceptably safe for a given application in a given environment. In GSN, the key elements are:
- **Goal (G)**:  A goal represents a safety claim or objective that the system or process aims to achieve.
- **Context (C)**: Context elements provide background information or assumptions relevant to understanding the safety argument. They help to establish the scope, boundaries, and constraints of the safety case.
- **Strategy (S)**: A strategy represents a high-level approach or plan for achieving a safety goal. Strategies outline the broad steps or methods that will be employed to fulfill the safety objectives.
- **Assumption (A)**: Assumptions represent conditions or premises that are accepted as true without requiring further justification. They are used to simplify the safety argument by providing a basis for reasoning. Assumptions should be explicitly stated and documented to ensure transparency and traceability.
- **Justification (J)**: Justifications provide evidence or reasoning to support the validity of a safety claim. They explain why a particular goal, strategy, or solution is justified and should be accepted. Justifications may include references to standards, regulations, test results, analyses, expert opinions, etc.
- **Solution (E)**: A solution represents a specific action, measure, or artifact that contributes to achieving a safety goal. Solutions are concrete implementations of strategies and may include design elements, procedures, or technologies.

Connections between these elements are:
- **SupportedBy connections are**: goal-to-goal, goal-to-strategy, goal-to-solution, strategy-to-goal.
```

```
- **InContextOf connections are**: goal-to-context, goal-to-assumption, goal-to-justification, strategy-to-context, strategy-to-assumption, strategy-to-justification.

**XML Formatting:**
Each GSN element must be formatted in XML. For example:
- Top-Goal: ‘<goal id="G1" description="The system is acceptably safe">...</goal>‘
- Sub-Goal: ‘<goal id="G2" description="The system hazards are mitigated" type="SupportedBy">...</goal>‘
- Context: ‘<context id="C1" description="Operational environment" type="InContextOf" />‘
- Strategy: ‘<strategy id="S1" description="Decompose into sub-goals" type="SupportedBy">...</strategy>‘
- Assumption: ‘<assumption id="A1" description="All relevant hazards have been identified" type="InContextOf" />‘
- Justification: ‘<justification id="J1" description="Hazard analysis follows standard XYZ" type="InContextOf" />‘
- Solution: ‘<solution id="E1" description="Hazard analysis report" type="SupportedBy"/>‘


**here is an Example of a Complete Safety Case in XML:**


**Example of a Complete Safety Case in XML:**

 <goal id="G1" description="Collision Avoidance System is acceptably safe" type="SupportedBy">
   <context id="C1" description="System Overview: Uses sensors and AI to detect and prevent collisions" type="InContextOf"/>
   <context id="C2" description="Boundaries include LIDAR, radar, cameras, and AI algorithms" type="InContextOf"/>
   <strategy id="S1" description="Ensure safe vehicle operation" type="SupportedBy">
       <goal id="G2" description="Detect obstacles accurately" type="SupportedBy">
           <context id="C3" description="Sensor subsystem detects obstacles" type="InContextOf"/>
           <assumption id="A1" description="Sensors function correctly" type="InContextOf"/>
           <justification id="J1" description="Sensor accuracy verified in testing" type="InContextOf"/>
           <solution id="E1" description="Sensor testing documentation" type="SupportedBy"/>
       </goal>
       <goal id="G3" description="Perform emergency braking timely" type="SupportedBy">
           <context id="C4" description="Braking system prevents collisions" type="InContextOf"/>
```

```
            <assumption id="A2" description="Braking response time meets requirements" type="InContextOf"/>
            <justification id="J2" description="Braking response validated in simulations" type="InContextOf"/>
            <solution id="E2" description="Braking system test results" type="SupportedBy"/>
        </goal>
        <goal id="G4" description="Provide driver alerts" type="SupportedBy">
            <context id="C5" description="Alerts warn driver of potential collisions" type="InContextOf"/>
            <assumption id="A3" description="Alerts function within specified timeframes" type="InContextOf"/>
            <justification id="J3" description="Alert timing validated in testing" type="InContextOf"/>
            <solution id="E3" description="Alert system validation records" type="SupportedBy"/>
        </goal>
    </strategy>
</goal>


 **Another Example of a Complete Safety Case in XML:**

<goal id="G1" description="Control System is acceptably safe to operate" type="SupportedBy">
    <context id="C1" description="Operating Role and Context" type="InContextOf"/>
    <context id="C2" description="Control System Definition" type="InContextOf"/>
    <goal id="G2" description="All identified hazards have been eliminated or sufficiently mitigated" type="SupportedBy">
        <context id="C3" description="Tolerability targets (Ref Z)" type="InContextOf"/>
        <context id="C4" description="Hazards identified from FHA (Ref Y)" type="InContextOf"/>
        <strategy id="S1" description="Argument over each identified hazard" type="SupportedBy">
            <goal id="G4" description="Hazard H1 has been eliminated" type="SupportedBy">
                <solution id="E1" description="Formel Verification" type="SupportedBy"/>
            </goal>
            <goal id="G5" description="Probability of Hazard H2 occurring less than 1 * 10^-6 per year" type="SupportedBy">
                <solution id="E2" description="Fault Tree Analysis" type="SupportedBy"/>
            </goal>
            <goal id="G6" description="Probability of Hazard H3 occurring less than 1 * 10^-6 per year" type="SupportedBy">
```

```
                <solution id="E3" description="Fault Tree Analysis"
                    ↪ type="SupportedBy"/>
            </goal>
        </strategy>
    </goal>
    <goal id="G3" description="Software in the control system has been
        ↪ developed to SIL appropriate to hazards involved"
        ↪ type="SupportedBy">
        <context id="C4" description="Hazards identified from FHA (Ref Y)"
            ↪ type="InContextOf"/>
        <strategy id="S2" description="Argument over allocated SIL for
            ↪ Primary and Secondary elements" type="SupportedBy">
            <justification id="J1" description="SIL apportionment is correct
                ↪ and complete" type="InContextOf"/>
            <context id="C6" description="Identified software hazards"
                ↪ type="InContextOf"/>
            <goal id="G7" description="Primary Protection System Developed to
                ↪ SIL 4" type="SupportedBy">
                <solution id="E4" description="Process Evidence for SIL4"
                    ↪ type="SupportedBy"/>
            </goal>
            <goal id="G8" description="Secondary Protection System
                ↪ Development to SIL 2" type="SupportedBy">
                <solution id="E5" description="Process Evidence for SIL2"
                    ↪ type="SupportedBy"/>
            </goal>
        </strategy>
    </goal>
</goal>


<goal id="G1" description="The Swift UAS is acceptably safe"
    ↪ type="SupportedBy">
    <context id="C1" description="specified mission" type="InContextOf" />
    <context id="C2" description="Specified configuration" type="InContextOf"
        ↪ />
    <context id="C3" description="Weather conditions during operation"
        ↪ type="InContextOf" />
    <context id="C4" description="Range: location and site where operated"
        ↪ type="InContextOf" />
    <context id="C5" description="Definition of acceptable safety as per NPR
        ↪ 8715.5A" type="InContextOf" />
    <strategy id="S1" description="Argument of hazard mitigation"
        ↪ type="SupportedBy">
        <justification id="J1" description="Safety is demonstrated through
            ↪ the identification and mitigation of all applicable hazards"
            ↪ type="InContextOf" />

        <goal id="G2" description="All identified Swift USA hazards are
            ↪ acceptably mitigated" type="SupportedBy">
```

```
            <context id="C7" description="Definition of acceptable
                ↪ mitigation" type="InContextOf" />
            <context id="C8" description="Swift UAS Safety analysis: Hazard
                ↪ log containing list of identified hazards"
                ↪ type="InContextOf" />
            <assumption id="A3" description="All relevant Swift USA hazards
                ↪ have been acceptably identified through hazard analysis"
                ↪ type="InContextOf" />
            <strategy id="S2" description="Argument over interactions between
                ↪ subsystems of the Swift UAS" type="SupportedBy">
                <context id="C6" description="Physical architecture of the
                    ↪ Swift UAS" type="InContextOf" />
                <goal id="G8" description="Hazards posed by interactions
                    ↪ between the Swift UAS subsystems are mitigated"
                    ↪ type="SupportedBy">
                    <context id="C10" description="Subsystem integration and
                        ↪ interaction assessment" type="InContextOf" />
                    <assumption id="A4" description="All subsystems are
                        ↪ correctly integrated and tested"
                        ↪ type="InContextOf" />
                    <strategy id="S4" description="Argument over the testing
                        ↪ of subsystem interactions" type="SupportedBy">
                        <goal id="G9" description="Subsystem interaction
                            ↪ tests confirm no emergent hazards"
                            ↪ type="SupportedBy">
                            <context id="C11" description="Test reports on
                                ↪ subsystem interaction" type="InContextOf"
                                ↪ />
                            <assumption id="A5" description="All potential
                                ↪ interactions between subsystems have been
                                ↪ identified" type="InContextOf" />
                            <strategy id="S5" description="Analysis of test
                                ↪ data" type="SupportedBy">
                                <goal id="G10" description="Test data shows
                                    ↪ no unexpected interactions leading to
                                    ↪ hazards" type="SupportedBy">
                                    <context id="C12" description="Data
                                        ↪ analysis methods and results"
                                        ↪ type="InContextOf" />
                                    <solution id="E1" description="Subsystem
                                        ↪ interaction analysis report
                                        ↪ confirming no unexpected hazards"
                                        ↪ type="Supports" />
                                </goal>
                            </strategy>
                        </goal>
                        <goal id="G11" description="Subsystems are designed
                            ↪ to minimize harmful interactions"
                            ↪ type="SupportedBy">
```

```
                            <context id="C13" description="Subsystem design
                                ↪ principles" type="InContextOf" />
                            <assumption id="A6" description="Design
                                ↪ principles have been followed"
                                ↪ type="InContextOf" />
                            <strategy id="S6" description="Design review and
                                ↪ verification" type="SupportedBy">
                                <goal id="G12" description="Design
                                    ↪ verification shows compliance with
                                    ↪ interaction guidelines"
                                    ↪ type="SupportedBy">
                                    <context id="C14" description="Design
                                        ↪ verification reports"
                                        ↪ type="InContextOf" />
                                    <solution id="E2" description="Design
                                        ↪ verification report showing
                                        ↪ adherence to interaction
                                        ↪ guidelines" type="Supports" />
                                </goal>
                            </strategy>
                        </goal>
                    </strategy>
                </goal>
            </strategy>
            <strategy id="S3" description="Argument over the physical
                ↪ architecture (breakdown over subsystems)"
                ↪ type="SupportedBy">
                <context id="C6" description="Physical architecture of the
                    ↪ Swift UAS" type="InContextOf" />
                <context id="C9" description="Swift UAS Design Management
                    ↪ Plan and Design Documentation" type="InContextOf" />
                <justification id="J2" description="The functional breakdown
                    ↪ of the Swift UAS mirrors the physical architecture.
                    ↪ i.e, for each function, there is a corresponding
                    ↪ physical system" type="InContextOf" />
                <goal id="G3" description="Hazards posed by the Swift UAS
                    ↪ ground control station (GCS) are mitigated"
                    ↪ type="SupportedBy">
                    <context id="C15" description="Ground Control Station
                        ↪ (GCS) configuration and operation"
                        ↪ type="InContextOf" />
                    <assumption id="A7" description="GCS hazards have been
                        ↪ fully identified and assessed" type="InContextOf"
                        ↪ />
                    <strategy id="S7" description="Argument over the safety
                        ↪ of GCS operation" type="SupportedBy">
                        <goal id="G13" description="GCS operation under all
                            ↪ conditions is demonstrated to be safe"
                            ↪ type="SupportedBy">
```

```
                            <context id="C16" description="GCS operational
                                ↪ scenarios" type="InContextOf" />
                            <assumption id="A8" description="All operational
                                ↪ scenarios have been tested"
                                ↪ type="InContextOf" />
                            <strategy id="S8" description="Testing of GCS
                                ↪ under various conditions"
                                ↪ type="SupportedBy">
                                <goal id="G14" description="Test results
                                    ↪ confirm GCS safety" type="SupportedBy">
                                    <context id="C17" description="GCS test
                                        ↪ results" type="InContextOf" />
                                    <solution id="E3" description="Test
                                        ↪ report showing GCS operation is
                                        ↪ safe under all scenarios"
                                        ↪ type="Supports" />
                                </goal>
                            </strategy>
                        </goal>
                        <goal id="G15" description="GCS design minimizes risk
                            ↪ of operational errors" type="SupportedBy">
                            <context id="C18" description="GCS design
                                ↪ principles and ergonomics"
                                ↪ type="InContextOf" />
                            <strategy id="S9" description="Review of GCS
                                ↪ design for safety" type="SupportedBy">
                                <goal id="G16" description="GCS design review
                                    ↪ confirms safety principles are
                                    ↪ applied" type="SupportedBy">
                                    <context id="C19" description="GCS design
                                        ↪ review documentation"
                                        ↪ type="InContextOf" />
                                    <solution id="E4" description="Design
                                        ↪ review report confirming GCS
                                        ↪ adheres to safety principles"
                                        ↪ type="Supports" />
                                </goal>
                            </strategy>
                        </goal>
                    </strategy>
                </goal>
                <goal id="G4" description="Hazards posed by the Swift UAS
                    ↪ communication infrastructure are mitigated"
                    ↪ type="SupportedBy">
                    <context id="C20" description="Communication
                        ↪ infrastructure details" type="InContextOf" />
                    <assumption id="A9" description="All
                        ↪ communication-related hazards have been
                        ↪ identified" type="InContextOf" />
```

```
                    <strategy id="S10" description="Argument over
                        ↪ communication reliability and security"
                        ↪ type="SupportedBy">
                        <goal id="G17" description="Communication channels
                            ↪ are reliable under all conditions"
                            ↪ type="SupportedBy">
                            <context id="C21" description="Communication
                                ↪ system test results" type="InContextOf" />
                            <strategy id="S11" description="Testing of
                                ↪ communication reliability"
                                ↪ type="SupportedBy">
                                <goal id="G18" description="Test results
                                    ↪ confirm reliability of communication
                                    ↪ systems" type="SupportedBy">
                                    <context id="C22" description="Detailed
                                        ↪ communication test data"
                                        ↪ type="InContextOf" />
                                    <solution id="E5" description="Test
                                        ↪ report confirming reliability of
                                        ↪ communication channels under all
                                        ↪ conditions" type="Supports" />
                                </goal>
                            </strategy>
                        </goal>
                        <goal id="G19" description="Communication
                            ↪ infrastructure is secure from unauthorized
                            ↪ access" type="SupportedBy">
                            <context id="C23" description="Security protocols
                                ↪ and measures" type="InContextOf" />
                            <strategy id="S12" description="Review of
                                ↪ security protocols" type="SupportedBy">
                                <goal id="G20" description="Security
                                    ↪ protocols review confirms robustness
                                    ↪ against threats" type="SupportedBy">
                                    <context id="C24" description="Security
                                        ↪ review documentation"
                                        ↪ type="InContextOf" />
                                    <solution id="E6" description="Security
                                        ↪ review report confirming
                                        ↪ robustness against unauthorized
                                        ↪ access and threats"
                                        ↪ type="Supports" />
                                </goal>
                            </strategy>
                        </goal>
                    </strategy>
                </goal>
            </strategy>
        </goal>
    </strategy>
```

```
</goal>



 **Here is another Example of a Complete Safety Case in XML:**

<goal id="G6" description="Behavior during descent of Autopilot module is correct" type="SupportedBy">
    <context id="C6" description="Swift UA Software Requirements" type="InContextOf" />
    <context id="C5" description="Swift UAV System Requirements" type="InContextOf" />
    <context id="C7" description="Definition of correctness: Implementation meets the informal specification describing the Autopilot module requirements" />
    <assumption id="A1" description="The requirements specification for Autopilot module are valid, complete and consistent" type="InContextOf" />

    <strategy id="S3" description="Argument that Autopilot module satisfies higher level requirements" type="SupportedBy">
        <goal id="G23" description="Autopilot system integration is complete and correct" type="SupportedBy">
            <strategy id="S14" description="Argument by testing integration of Autopilot module with other systems" type="SupportedBy">
                <context id="C16" description="Integration test plan for Autopilot module" type="InContextOf" />
                <goal id="G24" description="Integration test results show correct behavior of Autopilot during descent" type="SupportedBy">
                    <solution id="E5" description="Test results from integration tests of Autopilot module" type="SupportedBy" />
                </goal>
            </strategy>
        </goal>
    </strategy>

    <strategy id="S13" description="Argument of valid representation of aircraft data" type="SupportedBy">
        <goal id="G22" description="Aircraft state variables represent valid aircraft data" type="SupportedBy">
            <strategy id="S15" description="Argument by validation against sensor specifications and data consistency checks" type="SupportedBy">
                <context id="C17" description="Sensor data specifications and validation procedures" type="InContextOf" />
                <goal id="G25" description="Validation of sensor data against specifications and consistency checks is successful" type="SupportedBy">
```

```
                    <solution id="E6" description="Results from validation procedures on sensor data" type="SupportedBy" />
                </goal>
            </strategy>
        </goal>
    </strategy>

    <strategy id="S2" description="Argument of correct implementation over all sub-modules" type="SupportedBy">
        <context id="C4" description="Specification of Autopilot module" type="InContextOf" />
        <context id="C1" description="Autopilot module Software application" type="InContextOf" />

        <goal id="G4" description="FMS class implementation is correct" type="SupportedBy">
            <strategy id="S12" description="Argument by formalization and proof using AutoCert" type="SupportedBy">
                <goal id="G26" description="Formal verification of FMS implementation is correct" type="SupportedBy">
                    <context id="C18" description="Formal specification of FMS class" type="InContextOf" />
                    <solution id="E7" description="Proof of correct FMS implementation generated by AutoCert" type="SupportedBy" />
                </goal>
            </strategy>
        </goal>

        <goal id="G3" description="autopilot class implementation is correct" type="SupportedBy">
            <strategy id="S4" description="Argument over autopilot functions" type="SupportedBy">
                <goal id="G8" description="Computation of angle of attack is correctly implemented" type="SupportedBy">
                    <strategy id="S5" description="Argument by proof of correctness of implementation" type="SupportedBy">
                        <context id="C9" description="Specification for computing angle of attack" type="InContextOf" />
                        <goal id="G9" description="Proof of correct implementation of angle of attack computation generated using AutoCert verification tool" type="SupportedBy">
                            <context id="C8" description="AutoCert verification tool" type="InContextOf" />
                            <context id="C10" description="Automatic theorem prover" type="InContextOf" />
```

```
                            <solution id="E2" description="Proof of correct
                                ↪ implementation of computation of angle of
                                ↪ attack" type="SupportedBy" />
                        </goal>
                    </strategy>
                </goal>
            </strategy>
        </goal>
    </strategy>

    <strategy id="S1" description="Argument of valid specification over all
        ↪ sub-modules" type="SupportedBy">
        <context id="C3" description="Specification of autopilot module"
            ↪ type="InContextOf" />
        <context id="C1" description="Autopilot module software architecture"
            ↪ type="InContextOf" />
        <context id="C2" description="Autopilot control theory"
            ↪ type="InContextOf" />

        <goal id="G1" description="FMS specification is valid"
            ↪ type="SupportedBy">
            <strategy id="S16" description="Argument by validation of FMS
                ↪ requirements against system-level requirements"
                ↪ type="SupportedBy">
                <context id="C19" description="System-level requirements for
                    ↪ FMS" type="InContextOf" />
                <goal id="G27" description="Validation of FMS requirements
                    ↪ against system-level requirements is successful"
                    ↪ type="SupportedBy">
                    <solution id="E8" description="Validation results of FMS
                        ↪ requirements" type="SupportedBy" />
                </goal>
            </strategy>
        </goal>

        <goal id="G5" description="FMS specification is valid"
            ↪ type="SupportedBy">
            <strategy id="S6" description="Argument over breakdown of AP
                ↪ functionality" type="SupportedBy">
                <context id="C11" description="AP class design documentation"
                    ↪ type="InContextOf" />

                <goal id="G10" description="Specification of PID controller
                    ↪ updates for each aircraft controller is valid"
                    ↪ type="SupportedBy">
                    <strategy id="S17" description="Argument by review and
                        ↪ analysis of PID controller update process"
                        ↪ type="SupportedBy">
                        <context id="C20" description="PID controller design
                            ↪ documentation" type="InContextOf" />
```

```
                        <goal id="G28" description="Review and analysis
                            ↪ confirm that PID controller updates are
                            ↪ correctly specified" type="SupportedBy">
                            <solution id="E9" description="Review outcome and
                                ↪ analysis data" type="SupportedBy" />
                        </goal>
                    </strategy>
                </goal>

                <goal id="G11" description="Specification of the
                    ↪ initialization of AP object is valid"
                    ↪ type="SupportedBy">
                    <strategy id="S18" description="Argument by verification
                        ↪ of initialization procedures" type="SupportedBy">
                        <context id="C21" description="AP initialization
                            ↪ procedures documentation" type="InContextOf" />
                        <goal id="G29" description="Verification of AP object
                            ↪ initialization is successful"
                            ↪ type="SupportedBy">
                            <solution id="E10" description="Verification
                                ↪ results for AP object initialization"
                                ↪ type="SupportedBy" />
                        </goal>
                    </strategy>
                </goal>

                <goal id="G12" description="Specification of the
                    ↪ initialization of PID controller objects is valid"
                    ↪ type="SupportedBy">
                    <strategy id="S19" description="Argument by review and
                        ↪ simulation of PID controller initialization"
                        ↪ type="SupportedBy">
                        <context id="C22" description="Simulation results and
                            ↪ review of PID initialization process"
                            ↪ type="InContextOf" />
                        <goal id="G30" description="PID controller
                            ↪ initialization process is reviewed and
                            ↪ simulated successfully" type="SupportedBy">
                            <solution id="E11" description="Outcome of
                                ↪ simulation and review" type="SupportedBy"
                                ↪ />
                        </goal>
                    </strategy>
                </goal>
            </strategy>
        </goal>

        <goal id="G2" description="Autopilot specification is valid"
            ↪ type="SupportedBy">
```

```
        <strategy id="S7" description="Argument over breakdown of
            ↪ autopilot functionality" type="SupportedBy">
            <context id="C12" description="Autopilot class design
                ↪ documentation" type="InContextOf" />

            <goal id="G14" description="Specification of the
                ↪ initialization of FMS object is valid"
                ↪ type="SupportedBy">
                <strategy id="S20" description="Argument by verification
                    ↪ and validation of FMS object initialization"
                    ↪ type="SupportedBy">
                    <context id="C23" description="FMS object
                        ↪ initialization procedures" type="InContextOf"
                        ↪ />
                    <goal id="G31" description="FMS object initialization
                        ↪ is verified and validated successfully"
                        ↪ type="SupportedBy">
                        <solution id="E12" description="Verification and
                            ↪ validation results for FMS object
                            ↪ initialization" type="SupportedBy" />
                    </goal>
                </strategy>
            </goal>

            <goal id="G16" description="Specification of the reception of
                ↪ aircraft state information from sensors is valid"
                ↪ type="SupportedBy">
                <strategy id="S21" description="Argument by sensor data
                    ↪ validation and cross-checking" type="SupportedBy">
                    <context id="C24" description="Sensor data
                        ↪ cross-checking and validation procedures"
                        ↪ type="InContextOf" />
                    <goal id="G32" description="Sensor data reception and
                        ↪ validation is successful" type="SupportedBy">
                        <solution id="E13" description="Validation
                            ↪ results of sensor data reception"
                            ↪ type="SupportedBy" />
                    </goal>
                </strategy>
            </goal>

            <goal id="G17" description="Specification of the definition
                ↪ of current, previous and next waypoints is valid"
                ↪ type="SupportedBy">
                <strategy id="S22" description="Argument by validation of
                    ↪ waypoint data consistency and accuracy"
                    ↪ type="SupportedBy">
                    <context id="C25" description="Waypoint data
                        ↪ consistency and accuracy procedures"
                        ↪ type="InContextOf" />
```

```
                        <goal id="G33" description="Waypoint data is validated for consistency and accuracy" type="SupportedBy">
                            <solution id="E14" description="Validation results for waypoint data" type="SupportedBy" />
                        </goal>
                    </strategy>
                </goal>

                <goal id="G15" description="Specification of the computation of angle of attack is valid" type="SupportedBy">
                    <strategy id="S8" description="Argument that the correct formula is used in the specification" type="SupportedBy">
                        <goal id="G18" description="The specification uses the correct formula for computing angle of attack" type="SupportedBy">
                            <strategy id="S10" description="Argument by review (appeal to domain expertise)" type="SupportedBy">
                                <goal id="G20" description="The specification for computing angle of attack is reviewed to be correct by aircraft design team" type="SupportedBy">
                                    <context id="C15" description="Value of calibration parameter for pilot probe" type="InContextOf" />
                                    <context id="C14" description="Aircraft design team" type="InContextOf" />
                                    <solution id="E4" description="Outcome of review and review data" type="SupportedBy" />
                                </goal>
                            </strategy>
                        </goal>
                    </strategy>

                    <strategy id="S9" description="Argument that correct calibration constant is used in the specification" type="SupportedBy">
                        <goal id="G19" description="The calibration constant used in the specification is accurate" type="SupportedBy">
                            <strategy id="S11" description="Argument of correct experimental calibration" type="SupportedBy">
                                <goal id="G21" description="Pilot probe calibration is accurate" type="SupportedBy">
```

```
                                    <context id="C13" description="Wind
                                        ↪ tunnel experiments for air data
                                        ↪ probe" type="InContextOf" />
                                    <solution id="E3" description="Data from
                                        ↪ wind tunnel experiments on air
                                        ↪ data probe" type="SupportedBy" />
                                </goal>
                            </strategy>
                        </goal>
                    </strategy>
                </goal>
            </strategy>
        </goal>
    </strategy>
</goal>


**Here is another example of safety cases:**
<goal id="G1" description="Safe and reliable operation of Spacecraft
    ↪ Guidance, Navigation, and Control System (GNCS) across all mission
    ↪ phases, ensuring the system meets specified functional, performance,
    ↪ and safety requirements while mitigating all critical risks and
    ↪ hazards" type="SupportedBy">
    <context id="C1" description="System-Level Requirements for GNCS,
        ↪ encompassing all mission phases including pre-launch, ascent,
        ↪ orbital insertion, orbital operations, descent, re-entry, and
        ↪ landing" type="InContextOf" />
    <context id="C2" description="Mission Objectives and Requirements,
        ↪ derived from spacecraft design, intended mission duration, orbital
        ↪ characteristics, and environmental conditions" type="InContextOf"
        ↪ />
    <context id="C3" description="Definition of safe and reliable operation:
        ↪ All system components must perform within acceptable limits and
        ↪ thresholds during all mission phases, accounting for both normal
        ↪ operation and abnormal conditions such as component failures or
        ↪ external disturbances" type="InContextOf" />
    <assumption id="A1" description="The system-level requirements for the
        ↪ GNCS have been validated through stakeholder review and are
        ↪ consistent with the overall mission objectives and spacecraft
        ↪ design" type="InContextOf" />
    <assumption id="A2" description="All environmental models used to design
        ↪ and validate GNCS algorithms, including orbital mechanics,
        ↪ atmospheric dynamics, and space environment factors, accurately
        ↪ represent real-world conditions expected during the mission"
        ↪ type="InContextOf" />
    <assumption id="A3" description="All software and hardware components
        ↪ used in the GNCS have been independently tested and verified for
        ↪ integration, performance, and fault tolerance under a range of
        ↪ operational scenarios, including degraded modes and off-nominal
        ↪ conditions" type="InContextOf" />
```

```

    <strategy id="S1" description="Argument by decomposition of GNCS functionality into three core modules: Guidance, Navigation, and Control, with each module further decomposed into sub-functions, validation strategies, and fault mitigation mechanisms" type="SupportedBy">

        <goal id="G2" description="Correct and complete implementation of the Guidance Module, ensuring accurate trajectory planning, guidance parameter generation, and trajectory correction during all mission phases" type="SupportedBy">
            <context id="C4" description="Guidance Module is responsible for generating and updating the spacecraft’s desired trajectory based on mission objectives, spacecraft state, and environmental factors, accounting for uncertainties and external disturbances" type="InContextOf" />
            <context id="C5" description="Guidance algorithms must generate optimal trajectories that meet mission-specific constraints, such as fuel efficiency, time to target, and avoidance of hazardous regions of space, including debris fields and radiation zones" type="InContextOf" />
            <context id="C6" description="Assumption: Guidance algorithms have been verified using high-fidelity simulations that accurately model the spacecraft dynamics and environmental conditions across the entire mission profile" type="InContextOf" />
            <assumption id="A4" description="All sensor data used by the guidance system, including position, velocity, and orientation measurements, are accurate and reliable under nominal and off-nominal conditions" type="InContextOf" />

            <strategy id="S2" description="Argument by functional validation of guidance algorithms, including trajectory generation, trajectory correction, and contingency planning for off-nominal events such as sensor failure or navigation system errors" type="SupportedBy">
                <context id="C7" description="Guidance Algorithm Specification: Defines the algorithms used to generate the spacecraft’s desired trajectory based on mission objectives, spacecraft state, and environmental factors" type="InContextOf" />
                <goal id="G3" description="Guidance algorithms accurately generate optimal trajectories for all mission phases, including ascent, orbital operations, and re-entry" type="SupportedBy">
                    <strategy id="S3" description="Argument by simulation, formal analysis, and on-orbit validation of trajectory generation" type="SupportedBy">
                        <context id="C8" description="Trajectory generation algorithms have been validated using
```

```
    ↪ simulations that account for uncertainties in
    ↪ spacecraft state estimation, environmental
    ↪ disturbances, and system delays"
    ↪ type="InContextOf" />

<goal id="G4" description="Simulation results confirm
    ↪ that the trajectory generation algorithms
    ↪ consistently generate accurate trajectories
    ↪ under all expected mission conditions"
    ↪ type="SupportedBy">
    <context id="C24" description="Simulations
        ↪ include a comprehensive set of test cases,
        ↪ accounting for various mission phases and
        ↪ potential anomalies to ensure the
        ↪ robustness of trajectory generation"
        ↪ type="InContextOf" />
    <assumption id="A7" description="Simulation
        ↪ models accurately represent the dynamics
        ↪ of the spacecraft and the influence of
        ↪ environmental factors on trajectory
        ↪ generation" type="InContextOf" />
    <justification id="J1" description="Demonstrating
        ↪ consistent accuracy across simulated
        ↪ conditions builds confidence in the
        ↪ algorithms' reliability for real-world
        ↪ application." />
    <solution id="E1" description="Simulation results
        ↪ demonstrating the accuracy and reliability
        ↪ of the trajectory generation algorithms
        ↪ across a range of mission profiles,
        ↪ including nominal and off-nominal
        ↪ scenarios" type="SupportedBy" />
</goal>

<goal id="G5" description="Formal analysis of
    ↪ trajectory generation algorithms confirms
    ↪ mathematical correctness and robustness
    ↪ against input uncertainties"
    ↪ type="SupportedBy">
    <context id="C9" description="Formal methods have
        ↪ been applied to verify the mathematical
        ↪ correctness of the trajectory generation
        ↪ algorithms, including handling of
        ↪ uncertainties and external disturbances"
        ↪ type="InContextOf" />
    <assumption id="A8" description="The mathematical
        ↪ models used for formal verification are
        ↪ well-defined and cover all relevant
        ↪ aspects of the trajectory generation
        ↪ algorithms" type="InContextOf" />
```

```
                        <justification id="J2" description="Formal
                            ↪ analysis provides a high level of
                            ↪ assurance that algorithms can withstand
                            ↪ variations in input without failure." />
                        <solution id="E2" description="Formal
                            ↪ verification report for trajectory
                            ↪ generation algorithms, confirming their
                            ↪ correctness and robustness against
                            ↪ uncertainties and external disturbances"
                            ↪ type="SupportedBy" />
                    </goal>
                </strategy>
            </goal>

            <goal id="G6" description="Guidance system can accurately
                ↪ calculate and update guidance parameters based on
                ↪ real-time spacecraft state and environmental data"
                ↪ type="SupportedBy">
                <strategy id="S4" description="Argument by validation of
                    ↪ guidance parameter calculations against mission
                    ↪ profile data and real-time spacecraft state
                    ↪ measurements" type="SupportedBy">
                    <context id="C10" description="Mission profile data
                        ↪ includes expected spacecraft state at each
                        ↪ mission phase, including position, velocity,
                        ↪ orientation, and fuel usage"
                        ↪ type="InContextOf" />
                    <goal id="G7" description="Validation results confirm
                        ↪ that guidance parameters are calculated
                        ↪ correctly based on mission profile data and
                        ↪ real-time spacecraft state measurements"
                        ↪ type="SupportedBy">
                        <solution id="E3" description="Validation results
                            ↪ showing accurate calculation of guidance
                            ↪ parameters during various mission phases,
                            ↪ including ascent, orbital operations, and
                            ↪ re-entry" type="SupportedBy" />
                    </goal>
                </strategy>
            </goal>

            <goal id="G8" description="The guidance system can perform
                ↪ real-time trajectory correction in response to
                ↪ deviations from the desired trajectory or changes in
                ↪ mission objectives" type="SupportedBy">
                <strategy id="S5" description="Argument by validation of
                    ↪ real-time trajectory correction algorithms"
                    ↪ type="SupportedBy">
                    <context id="C11" description="Trajectory correction
                        ↪ algorithms must adjust the spacecraft's
```

```
                                ↪ current trajectory to bring it back onto the
                                ↪ desired path while minimizing fuel usage and
                                ↪ avoiding hazardous areas" type="InContextOf" />
                        <goal id="G9" description="Real-time trajectory
                            ↪ correction is performed accurately and
                            ↪ efficiently under all expected mission
                            ↪ scenarios" type="SupportedBy">
                            <context id="C25" description="Real-time
                                ↪ performance metrics will be collected
                                ↪ during trajectory correction scenarios to
                                ↪ ensure compliance with mission parameters"
                                ↪ type="InContextOf" />
                            <assumption id="A9" description="Real-time data
                                ↪ processing capabilities of the guidance
                                ↪ system are sufficient to support rapid
                                ↪ trajectory adjustments under dynamic
                                ↪ conditions" type="InContextOf" />
                            <justification id="J3" description="Efficient
                                ↪ real-time trajectory corrections ensure
                                ↪ mission success and prevent deviations
                                ↪ that could lead to system failures." />
                            <solution id="E4" description="Test results from
                                ↪ simulations and on-orbit validation,
                                ↪ showing the successful correction of
                                ↪ trajectory deviations under various
                                ↪ mission conditions" type="SupportedBy" />
                        </goal>
                    </strategy>
                </goal>
            </strategy>
        </goal>

        <goal id="G10" description="Correct and complete implementation of
            ↪ the Navigation Module, ensuring accurate and reliable state
            ↪ estimation based on sensor data, environmental models, and
            ↪ spacecraft dynamics" type="SupportedBy">
            <context id="C12" description="The Navigation Module is
                ↪ responsible for determining the spacecraft's position,
                ↪ velocity, and orientation in real-time using data from
                ↪ onboard sensors and external reference sources, such as
                ↪ GPS, star trackers, and inertial measurement units (IMUs)"
                ↪ type="InContextOf" />
            <context id="C13" description="Navigation algorithms must account
                ↪ for sensor noise, delays, and inaccuracies, as well as
                ↪ uncertainties in the spacecraft's dynamics and
                ↪ environmental models" type="InContextOf" />
            <assumption id="A5" description="All sensors used by the
                ↪ navigation system, including GPS, star trackers, and IMUs,
                ↪ provide accurate and reliable data under nominal and
                ↪ off-nominal conditions" type="InContextOf" />
```

```

        <strategy id="S6" description="Argument by functional validation
            ↪ of navigation algorithms, including state estimation,
            ↪ sensor fusion, and fault detection and recovery"
            ↪ type="SupportedBy">
            <context id="C14" description="Navigation algorithms must
                ↪ accurately estimate the spacecraft’s state using data
                ↪ from multiple sensors, while also detecting and
                ↪ compensating for sensor faults and errors"
                ↪ type="InContextOf" />
            <goal id="G11" description="Navigation algorithms provide
                ↪ accurate state estimates within specified tolerances
                ↪ under all expected mission conditions"
                ↪ type="SupportedBy">
                <strategy id="S7" description="Argument by simulation,
                    ↪ formal analysis, and on-orbit validation of state
                    ↪ estimation algorithms" type="SupportedBy">
                    <goal id="G12" description="Simulation results
                        ↪ confirm that state estimation algorithms
                        ↪ consistently provide accurate estimates under
                        ↪ varying sensor conditions and environmental
                        ↪ influences" type="SupportedBy">
                        <context id="C15" description="Simulations
                            ↪ include various scenarios with sensor
                            ↪ noise, delays, and inaccuracies to ensure
                            ↪ robustness of state estimation algorithms"
                            ↪ type="InContextOf" />
                        <assumption id="A6" description="Simulation
                            ↪ models accurately represent the dynamics
                            ↪ of the spacecraft and the influence of
                            ↪ environmental factors on sensor
                            ↪ measurements" type="InContextOf" />
                        <justification id="J4" description="Demonstrating
                            ↪ consistent accuracy across simulated
                            ↪ conditions builds confidence in the
                            ↪ algorithms’ reliability for real-world
                            ↪ application." />
                        <solution id="E5" description="Simulation results
                            ↪ demonstrating the accuracy and reliability
                            ↪ of the state estimation algorithms across
                            ↪ a range of mission profiles, including
                            ↪ nominal and off-nominal scenarios"
                            ↪ type="SupportedBy" />
                    </goal>

                    <goal id="G13" description="Formal analysis of state
                        ↪ estimation algorithms confirms mathematical
                        ↪ correctness and robustness against input
                        ↪ uncertainties" type="SupportedBy">
```

```
                    <context id="C16" description="Formal methods
                        ↪ have been applied to verify the
                        ↪ mathematical correctness of the state
                        ↪ estimation algorithms, including handling
                        ↪ of uncertainties and external
                        ↪ disturbances" type="InContextOf" />
                    <assumption id="A10" description="The
                        ↪ mathematical models used for formal
                        ↪ verification are well-defined and cover
                        ↪ all relevant aspects of the state
                        ↪ estimation algorithms" type="InContextOf"
                        ↪ />
                    <justification id="J5" description="Formal
                        ↪ analysis provides a high level of
                        ↪ assurance that algorithms can withstand
                        ↪ variations in input without failure." />
                    <solution id="E6" description="Formal
                        ↪ verification report for state estimation
                        ↪ algorithms, confirming their correctness
                        ↪ and robustness against uncertainties and
                        ↪ external disturbances" type="SupportedBy"
                        ↪ />
                </goal>
            </strategy>
        </goal>

        <goal id="G14" description="The navigation system can
            ↪ accurately calculate and update state estimates based
            ↪ on real-time sensor data and environmental models"
            ↪ type="SupportedBy">
            <strategy id="S8" description="Argument by validation of
                ↪ state estimate calculations against mission
                ↪ profile data and real-time sensor measurements"
                ↪ type="SupportedBy">
                <goal id="G15" description="Validation results
                    ↪ confirm that state estimates are calculated
                    ↪ correctly based on real-time sensor data and
                    ↪ environmental models" type="SupportedBy">
                    <solution id="E7" description="Validation results
                        ↪ showing accurate calculation of state
                        ↪ estimates during various mission phases,
                        ↪ including ascent, orbital operations, and
                        ↪ re-entry" type="SupportedBy" />
                </goal>
            </strategy>
        </goal>

        <goal id="G16" description="The navigation system can detect
            ↪ and recover from sensor faults to maintain accurate
            ↪ state estimation" type="SupportedBy">
```

```
                <strategy id="S9" description="Argument by validation of
                    ↪ fault detection and recovery algorithms in
                    ↪ navigation systems" type="SupportedBy">
                    <goal id="G17" description="Fault detection
                        ↪ algorithms accurately identify sensor faults,
                        ↪ enabling timely recovery and maintaining state
                        ↪ estimation accuracy" type="SupportedBy">
                        <context id="C17" description="Fault detection
                            ↪ algorithms must monitor sensor data in
                            ↪ real-time, identifying anomalies and
                            ↪ enabling recovery actions to maintain
                            ↪ accurate state estimation"
                            ↪ type="InContextOf" />
                        <assumption id="A11" description="Fault detection
                            ↪ algorithms are effective and can
                            ↪ differentiate between normal sensor
                            ↪ behavior and faulty readings under all
                            ↪ expected conditions" type="InContextOf" />
                        <justification id="J6" description="Timely
                            ↪ detection of faults prevents erroneous
                            ↪ state estimates and potential mission
                            ↪ failures." />
                        <solution id="E8" description="Test results from
                            ↪ simulations and on-orbit validation,
                            ↪ showing successful fault detection and
                            ↪ recovery under various scenarios"
                            ↪ type="SupportedBy" />
                    </goal>
                </strategy>
            </goal>
        </strategy>
    </goal>

    <goal id="G18" description="Correct and complete implementation of
        ↪ the Control Module, ensuring accurate and reliable control of
        ↪ spacecraft dynamics based on state estimates and guidance
        ↪ commands" type="SupportedBy">
        <context id="C18" description="Control Module is responsible for
            ↪ executing control commands to maintain the spacecraft’s
            ↪ desired trajectory and orientation, adjusting for
            ↪ disturbances and ensuring mission success"
            ↪ type="InContextOf" />
        <context id="C19" description="Control algorithms must account
            ↪ for spacecraft dynamics, including inertia, thrust
            ↪ dynamics, and external disturbances such as atmospheric
            ↪ drag and gravitational influences" type="InContextOf" />
        <assumption id="A12" description="All actuators and propulsion
            ↪ systems used by the control module provide accurate and
            ↪ reliable performance under nominal and off-nominal
            ↪ conditions" type="InContextOf" />
```

```

            <strategy id="S10" description="Argument by functional validation
                ↪ of control algorithms, including trajectory tracking,
                ↪ attitude control, and fault detection and recovery"
                ↪ type="SupportedBy">
                <goal id="G19" description="Control algorithms provide
                    ↪ accurate and reliable control of spacecraft dynamics
                    ↪ under all expected mission conditions"
                    ↪ type="SupportedBy">
                    <strategy id="S11" description="Argument by simulation,
                        ↪ formal analysis, and on-orbit validation of
                        ↪ control algorithms" type="SupportedBy">
                        <goal id="G20" description="Simulation results
                            ↪ confirm that control algorithms consistently
                            ↪ maintain spacecraft dynamics within specified
                            ↪ tolerances under varying conditions"
                            ↪ type="SupportedBy">
                            <context id="C20" description="Simulations
                                ↪ include various scenarios with
                                ↪ disturbances and uncertainties to ensure
                                ↪ robustness of control algorithms"
                                ↪ type="InContextOf" />
                            <assumption id="A13" description="Simulation
                                ↪ models accurately represent the dynamics
                                ↪ of the spacecraft and the influence of
                                ↪ environmental factors on control
                                ↪ performance" type="InContextOf" />
                            <justification id="J7" description="Demonstrating
                                ↪ consistent performance across simulated
                                ↪ conditions builds confidence in the
                                ↪ algorithms' reliability for real-world
                                ↪ application." />
                            <solution id="E9" description="Simulation results
                                ↪ demonstrating the accuracy and reliability
                                ↪ of the control algorithms across a range
                                ↪ of mission profiles, including nominal and
                                ↪ off-nominal scenarios" type="SupportedBy"
                                ↪ />
                        </goal>

                        <goal id="G21" description="Formal analysis of
                            ↪ control algorithms confirms mathematical
                            ↪ correctness and robustness against input
                            ↪ uncertainties" type="SupportedBy">
                            <context id="C21" description="Formal methods
                                ↪ have been applied to verify the
                                ↪ mathematical correctness of the control
                                ↪ algorithms, including handling of
                                ↪ uncertainties and external disturbances"
                                ↪ type="InContextOf" />
```

```
                            <assumption id="A14" description="The
                                ↪ mathematical models used for formal
                                ↪ verification are well-defined and cover
                                ↪ all relevant aspects of the control
                                ↪ algorithms" type="InContextOf" />
                            <justification id="J8" description="Formal
                                ↪ analysis provides a high level of
                                ↪ assurance that algorithms can withstand
                                ↪ variations in input without failure." />
                            <solution id="E10" description="Formal
                                ↪ verification report for control
                                ↪ algorithms, confirming their correctness
                                ↪ and robustness against uncertainties and
                                ↪ external disturbances" type="SupportedBy"
                                ↪ />
                        </goal>
                    </strategy>
                </goal>

                <goal id="G22" description="The control system can accurately
                    ↪ execute control commands based on real-time state
                    ↪ estimates and guidance commands" type="SupportedBy">
                    <strategy id="S12" description="Argument by validation of
                        ↪ control command execution against mission profile
                        ↪ data and real-time state estimates"
                        ↪ type="SupportedBy">
                        <goal id="G23" description="Validation results
                            ↪ confirm that control commands are executed
                            ↪ correctly based on real-time state estimates
                            ↪ and guidance commands" type="SupportedBy">
                            <solution id="E11" description="Validation
                                ↪ results showing accurate execution of
                                ↪ control commands during various mission
                                ↪ phases, including ascent, orbital
                                ↪ operations, and re-entry"
                                ↪ type="SupportedBy" />
                        </goal>
                    </strategy>
                </goal>

                <goal id="G24" description="The control system can detect and
                    ↪ recover from actuator faults to maintain accurate
                    ↪ control of spacecraft dynamics" type="SupportedBy">
                    <strategy id="S13" description="Argument by validation of
                        ↪ fault detection and recovery algorithms in control
                        ↪ systems" type="SupportedBy">
                        <goal id="G25" description="Fault detection
                            ↪ algorithms accurately identify actuator
                            ↪ faults, enabling timely recovery and
```

```
                            ↪ maintaining control performance"
                            ↪ type="SupportedBy">
                            <context id="C22" description="Fault detection
                                ↪ algorithms must monitor actuator
                                ↪ performance in real-time, identifying
                                ↪ anomalies and enabling recovery actions to
                                ↪ maintain control performance"
                                ↪ type="InContextOf" />
                            <assumption id="A15" description="Fault detection
                                ↪ algorithms are effective and can
                                ↪ differentiate between normal actuator
                                ↪ behavior and faulty performance under all
                                ↪ expected conditions" type="InContextOf" />
                            <justification id="J9" description="Timely
                                ↪ detection of faults prevents control
                                ↪ errors and potential mission failures." />
                            <solution id="E12" description="Test results from
                                ↪ simulations and on-orbit validation,
                                ↪ showing successful fault detection and
                                ↪ recovery under various scenarios"
                                ↪ type="SupportedBy" />
                        </goal>
                    </strategy>
                </goal>
            </strategy>
        </goal>

        <goal id="G26" description="Overall GNCS functionality is validated
            ↪ through system-level testing, ensuring that the integrated
            ↪ system meets all safety and performance requirements under all
            ↪ expected operational conditions" type="SupportedBy">
            <context id="C23" description="System-level testing includes
                ↪ integrated tests of the Guidance, Navigation, and Control
                ↪ modules, simulating all mission phases and operational
                ↪ scenarios" type="InContextOf" />
            <context id="C24" description="Test results will be documented
                ↪ and analyzed to ensure compliance with all safety and
                ↪ performance requirements" type="InContextOf" />
            <assumption id="A16" description="All components of the GNCS are
                ↪ designed and implemented according to established
                ↪ standards for safety, reliability, and performance"
                ↪ type="InContextOf" />
            <justification id="J10" description="Validating the integrated
                ↪ GNCS through comprehensive testing ensures mission success
                ↪ and the safety of all crew members and systems on board."
                ↪ />
            <solution id="E13" description="System-level test reports
                ↪ demonstrating successful integration and validation of the
                ↪ Guidance, Navigation, and Control modules under all
                ↪ expected conditions" type="SupportedBy" />
```

```
        </goal>
    </strategy>
</goal>


**Standards to Consider:**
Please ensure that the safety cases adhere to relevant safety standards such as ISO 26262, SOTIF, DO-178C, or other applicable safety standards, depending on the system domain.

**Output Requirements:**

- In output I want to have just one safety case that will constitute a single GSN diagram and not several safety cases that will make several different GSN diagrams.
- Please ensure that the output adhere to valid XML syntax. Use appropriate XML tags for each element of the GSN, with each tag enclosing relevant content (e.g.,  <goal>, <strategy>, <solution>, etc.)
- Generate well-formated XML code as the output for each safety case, starting by <goal> and ending by </goal>, without something thing before and after
- The safety cases must be detailed, precise, consistent, and persuasive to establish confidence in the system's safety.
- The generated content should be just the safety cases and nothing anymore, nothing, Nothing. I just need to see the safety cases Please
- Safety cases must have at least 5 levels.
- Please enrich the safety cases by using assumptions and justifications as far as possible at each goal, even on the lowest levels of the safety cases (level 5, level 6, level 7, level 8, etc.). Take into consideration the linguistic formulation of these GSN elements.
- Provide as much information as possible when generating the lower-level GSN elements (context, assumption, solution, justification, goals), and ensure that they are as detailed as the higher-level GSN elements.
- Please, I don't need partial safety cases, generate safety cases that are as complete as possible, and that don't need further revision by a specialist.
- The most important: The generated safety cases should have minimum 10000 Tokens
"""
```

## A.3 THE U-SHIFT ONBOARDING SYSTEM'S SAFETY ANALYSIS AND SYSTEM DESCRIPTION

Listing A.3: Safety Analysis and system description for the U-SHIFT Onboarding Subsystem

```

## System Description and Safety Analysis: U-SHIFT Onboarding System

```

```
---

#### 1. System Definition

**System Overview**:
The Onboarding System manages passenger access, including ramp deployment, door operations, and interaction with passenger control inputs. This system operates autonomously, ensuring smooth and barrier-free access to the capsule for all passengers, including those with mobility limitations. It is integrated with the vehicle's Control System to ensure safe and efficient operations during passenger boarding and disembarking.

**System Boundaries**:
- **Subsystems**:
  - **Ramp Control Subsystem**: Deploys and retracts the ramp for barrier-free access.
  - **Door Operation Subsystem**: Automates door opening and closing.
  - **Obstacle Detection Subsystem**: Detects obstacles near the doors and ramp to prevent unsafe deployment.
  - **Passenger Safety Interface**: Provides alerts and controls for safe passenger interaction.

- **Components**:
  - **Ramp Actuator**: Motor-driven mechanism for ramp deployment and retraction.
  - **Door Actuator**: Mechanism to securely open and close doors.
  - **Obstacle Sensors**: Proximity sensors to detect objects or people near the doors and ramp area.
  - **Control Interface**: Integrated with the Control System for coordinated boarding processes.

---

#### 2. Operational Environment

**Physical Environment**:
The Onboarding System is designed to operate at passenger pickup and drop-off locations, such as bus stops and designated loading zones. It is adaptable to various physical conditions:
- **Surface Types**: Flat, stable surfaces including asphalt and concrete.
- **Weather Conditions**: Operates in various weather conditions, with sensors detecting slippery surfaces for safe ramp deployment.
- **Lighting Conditions**: Operates effectively in day, night, and low-light conditions with sensor support.

**Operational Scenarios**:
- **Scenario 1**: Passenger onboarding at a bus stop with the ramp deployed for wheelchair access.
- **Scenario 2**: Automatic door closure after all passengers have boarded.
```

```
- **Scenario 3**: Preventing ramp deployment if an obstacle is detected.

**Human Interaction**:
Passengers and operators receive safety notifications and have the option to manually intervene:
- **Auditory and Visual Alerts**: Alert passengers of door and ramp movement.
- **Manual Override**: Operators can remotely control ramp and door functions in case of failure or emergency.

---

#### 3. Functional and Performance Requirements

**Functional Requirements**:
- **Ramp Deployment Safety**: Ramp should only deploy if no obstacles are detected.
- **Secure Door Operation**: Doors must remain closed and locked when the vehicle is in motion.
- **Passenger Alerts**: Issue alerts before and during ramp or door operations to inform passengers.

**Performance Requirements**:
- **Deployment Time**: Ramp deployment or retraction must complete within 5 seconds.
- **Detection Precision**: Obstacle sensors must detect objects within 0.5 meters with 95% accuracy.
- **Reliability**: The system should achieve 99% operational uptime.

---

#### 4. Safety-Critical Functions

**Identification**:
- **Function 1**: Safe deployment of ramp and prevention of deployment when obstacles are present.
- **Function 2**: Secure locking and unlocking of doors in sync with vehicle motion.
- **Function 3**: Provision of alerts for safe passenger onboarding and disembarking.

**Failure Modes**:
- **Mode 1**: Failure to retract the ramp, causing potential collision with ground obstacles.
- **Mode 2**: Unintended door operation, leading to possible safety risks.
- **Mode 3**: Obstacle sensor malfunction, failing to detect passengers or obstacles.

---

#### 5. System Architecture and Design
```

```

**Architecture Overview**:
The Onboarding System is built with modular components, each handling a specific function related to passenger safety and access. It integrates with the central Control System, allowing for coordinated operations with the Drive System and Communication System.

**Design Principles**:
- **Redundancy**: Key components, such as door actuators and obstacle sensors, have redundant systems to prevent failures.
- **Fail-safe Operation**: If an obstacle is detected, the system halts ramp and door operations to prevent accidents.

**Safety Mechanisms**:
- **Sensor Cross-validation**: Multiple sensors provide obstacle data to ensure accurate detection before ramp deployment.
- **Self-diagnostics**: Continuous monitoring of ramp and door actuators for early detection of faults.

---

#### 6. Implementation Details

**Software**:
The Onboarding System software includes:
1. **Ramp Control Module**: Controls ramp deployment and retraction based on obstacle sensor data.
2. **Door Control Module**: Manages door opening and closing, with safety locks for vehicle motion.
3. **Safety Alert Module**: Issues notifications to passengers during ramp or door operations.

**Hardware**:
- **Sensors**: Proximity sensors with short-range detection for accurate obstacle monitoring.
- **Processing Unit**: Manages decisions for ramp and door operation.
- **Actuators**: Motor-driven ramp and door actuators designed for reliable and quick response.

**Integration and Configuration Management**:
The Onboarding System is integrated with the U-SHIFT Control System for coordinated actions, particularly during stops. Configuration management includes regular software updates and version control for system consistency.

---

#### 7. Verification and Validation

**Testing Procedures**:
```

```
- **Unit Testing**: Tests individual components, such as ramp actuators and sensors, under controlled lab conditions.
- **Integration Testing**: Simulates full system operation, focusing on ramp and door safety during passenger boarding.
- **Environmental Testing**: Evaluates system response to different weather and surface conditions to ensure reliable operation.

**Validation Methods**:
- **Simulation Testing**: Uses simulated boarding scenarios to test ramp and door responses.
- **Field Testing**: Real-world testing at designated stops, with various passenger loads and obstacle configurations.

**Test Results**:
- Obstacle detection accuracy: 95% within specified range.
- Ramp deployment and retraction: Completed within 5 seconds in 98% of tests.

---

#### 8. Maintenance and Operation

**Maintenance Procedures**:
- **Sensor Calibration**: Regular calibration to maintain detection accuracy.
- **Actuator Inspection**: Periodic checks of ramp and door actuators for wear and tear.
- **Software Updates**: Regular updates to improve system performance and address any known issues.

**Operational Procedures**:
- The Onboarding System is active during vehicle stops and standby when in motion. Operators can remotely manage ramp and door functions.
- System logs are maintained for each onboarding event to assist with maintenance and troubleshooting.

**Incident Response**:
- Any unexpected system behavior, such as a failure to deploy or retract the ramp, generates an automatic incident report for diagnostic review.

---

#### 9. Regulatory and Compliance Information

**Applicable Standards**:
- **ISO 26262**: Functional safety for road vehicles.
- **UNECE R107**: Requirements for accessibility and passenger safety.
- **SAE J3016**: Standards for automation levels relevant to passenger boarding.

**Compliance Strategy**:
```

```
- The Onboarding System is tested and verified to meet regulatory requirements, with periodic audits to ensure compliance with relevant standards.

---

#### 10. Safety Analysis

**Purpose**:
Identify potential hazards in the Onboarding System, assess associated risks, and define mitigations to ensure safe operation for passenger access.

---

#### 10.1 Hazard Identification

| **ID**  | **Hazard**                         | **Description**                          |
|---------|------------------------------------|------------------------------------------------------------
| **H1**  | Ramp not retracted                 | Ramp fails to retract, posing a collision hazard.        |
| **H2**  | Unintended door opening            | Doors open unexpectedly, risking passenger safety.      |
| **H3**  | Obstacle not detected              | Obstacle sensors fail, leading to potential ramp deployment risks.|
| **H4**  | Delay in door or ramp operation    | Delay in operation may lead to passenger confusion or injury.   |

---

#### 10.2 Risk Assessment

| **Hazard ID** | **Likelihood** | **Severity** | **Risk Rating** | **Mitigation Strategy**                              |
|---------------|----------------|--------------|-----------------|------------------------------------------------------------
| **H1**        | Likely (3)     | Severe (4)   | High (12)       | Regular actuator maintenance and self-diagnostic checks.     |
| **H2**        | Occasional (2) | Severe (4)   | Medium (8)      | Door lock with motion-detection to prevent opening while moving. |
| **H3**        | Rare (1)       | Major (3)    | Medium (3)      | Cross-validation with multiple sensors.                      |
| **H4**        | Unlikely (1)   | Moderate (3) | Low (3)         | Timer-based monitoring and alerts for delayed operations.    |

---

#### 10.3 Risk Mitigation Strategies

- **H1 - Ramp Not Retracted**: Implement redundant actuator checks and safety locks to ensure reliable ramp retraction.
```

```
- **H2 - Unintended Door Opening**: Use vehicle motion locks to prevent door operation while in transit.
- **H3 - Obstacle Not Detected**: Enhance detection accuracy by using a network of sensors and cross-referencing data.
- **H4 - Delay in Operation**: Configure system alerts to inform passengers of any operational delays.
```

## A.4 THE U-SHIFT ONBOARDING SYSTEM'S SAFETY CASES IN XML BASED ON GSN

Listing A.4: Safety Cases for the U-SHIFT Onboarding system

```
<goal id="G1" description="The Onboarding System ensures safe passenger access, including ramp deployment, door operation, and obstacle detection" type="SupportedBy">
    <context id="C1" description="System Overview: The Onboarding System manages passenger access, including ramp deployment, door operations, and interaction with passenger control inputs" type="InContextOf" />
    <context id="C2" description="System Boundaries: The Onboarding System includes the Ramp Control Subsystem, Door Operation Subsystem, Obstacle Detection Subsystem, and Passenger Safety Interface" type="InContextOf" />
    <assumption id="A1" description="All subsystems are designed and implemented according to established standards for safety, reliability, and performance" type="InContextOf" />
    <justification id="J1" description="The system-level validation of the Onboarding System ensures that all components and interactions are safe and reliable, meeting or exceeding all applicable standards for passenger access" type="InContextOf" />
    <strategy id="S1" description="Argument by decomposition of the Onboarding System functionality into its core modules, each validated through formal analysis and testing" type="SupportedBy">
        <goal id="G2" description="The Ramp Control Subsystem ensures safe ramp deployment and retraction" type="SupportedBy">
            <context id="C3" description="Ramp Control Subsystem: Deploys and retracts the ramp for barrier-free access" type="InContextOf" />
            <assumption id="A2" description="The ramp actuator and sensors are designed and implemented according to established standards for safety, reliability, and performance" type="InContextOf" />
            <justification id="J2" description="The ramp control algorithm is designed to prevent ramp deployment if obstacles are detected within the specified range" type="InContextOf" />
            <strategy id="S2" description="Argument by functional validation of the ramp control algorithm, including tests for obstacle detection and ramp deployment/retraction" type="SupportedBy">
```

```
                <goal id="G3" description="The ramp control algorithm
                    ↪ correctly detects obstacles within the specified
                    ↪ range" type="SupportedBy">
                    <solution id="E1" description="Test results showing
                        ↪ successful obstacle detection under various
                        ↪ conditions" type="SupportedBy" />
                </goal>
                <goal id="G4" description="The ramp deployment/retraction is
                    ↪ executed correctly under all expected conditions"
                    ↪ type="SupportedBy">
                    <solution id="E2" description="Simulation results showing
                        ↪ successful ramp deployment/retraction under
                        ↪ various conditions" type="SupportedBy" />
                </goal>
            </strategy>
        </goal>
        <goal id="G5" description="The Door Operation Subsystem ensures safe
            ↪ door opening and closing" type="SupportedBy">
            <context id="C4" description="Door Operation Subsystem: Automates
                ↪ door opening and closing" type="InContextOf" />
            <assumption id="A3" description="The door actuators and sensors
                ↪ are designed and implemented according to established
                ↪ standards for safety, reliability, and performance"
                ↪ type="InContextOf" />
            <justification id="J3" description="The door control algorithm is
                ↪ designed to prevent door opening if the vehicle is in
                ↪ motion" type="InContextOf" />
            <strategy id="S3" description="Argument by functional validation
                ↪ of the door control algorithm, including tests for door
                ↪ opening/closing and vehicle motion detection"
                ↪ type="SupportedBy">
                <goal id="G6" description="The door control algorithm
                    ↪ correctly detects vehicle motion" type="SupportedBy">
                    <solution id="E3" description="Test results showing
                        ↪ successful detection of vehicle motion under
                        ↪ various conditions" type="SupportedBy" />
                </goal>
                <goal id="G7" description="The door opening/closing is
                    ↪ executed correctly under all expected conditions"
                    ↪ type="SupportedBy">
                    <solution id="E4" description="Simulation results showing
                        ↪ successful door opening/closing under various
                        ↪ conditions" type="SupportedBy" />
                </goal>
            </strategy>
        </goal>
        <goal id="G8" description="The Obstacle Detection Subsystem ensures
            ↪ safe obstacle detection" type="SupportedBy">
```

```
            <context id="C5" description="Obstacle Detection Subsystem:
                ↪ Detects obstacles near the doors and ramp"
                ↪ type="InContextOf" />
            <assumption id="A4" description="The obstacle sensors are
                ↪ designed and implemented according to established
                ↪ standards for safety, reliability, and performance"
                ↪ type="InContextOf" />
            <justification id="J4" description="The obstacle detection
                ↪ algorithm is designed to prevent ramp deployment if
                ↪ obstacles are detected within the specified range"
                ↪ type="InContextOf" />
            <strategy id="S4" description="Argument by functional validation
                ↪ of the obstacle detection algorithm, including tests for
                ↪ obstacle detection and ramp deployment prevention"
                ↪ type="SupportedBy">
                <goal id="G9" description="The obstacle detection algorithm
                    ↪ correctly detects obstacles within the specified
                    ↪ range" type="SupportedBy">
                    <solution id="E5" description="Test results showing
                        ↪ successful obstacle detection under various
                        ↪ conditions" type="SupportedBy" />
                </goal>
                <goal id="G10" description="The ramp deployment is prevented
                    ↪ if obstacles are detected within the specified range"
                    ↪ type="SupportedBy">
                    <solution id="E6" description="Simulation results showing
                        ↪ successful ramp deployment prevention under
                        ↪ various conditions" type="SupportedBy" />
                </goal>
            </strategy>
        </goal>
        <goal id="G11" description="The Passenger Safety Interface ensures
            ↪ safe passenger interaction" type="SupportedBy">
            <context id="C6" description="Passenger Safety Interface:
                ↪ Provides alerts and controls for safe passenger
                ↪ interaction" type="InContextOf" />
            <assumption id="A5" description="The passenger safety interface
                ↪ is designed and implemented according to established
                ↪ standards for safety, reliability, and performance"
                ↪ type="InContextOf" />
            <justification id="J5" description="The passenger safety
                ↪ interface is designed to provide clear and timely alerts
                ↪ to passengers" type="InContextOf" />
            <strategy id="S5" description="Argument by functional validation
                ↪ of the passenger safety interface, including tests for
                ↪ alert generation and passenger interaction"
                ↪ type="SupportedBy">
                <goal id="G12" description="The passenger safety interface
                    ↪ correctly generates alerts to passengers"
                    ↪ type="SupportedBy">
```

```
                    <solution id="E7" description="Test results showing successful alert generation under various conditions" type="SupportedBy" />
                </goal>
                <goal id="G13" description="The passenger interaction is executed correctly under all expected conditions" type="SupportedBy">
                    <solution id="E8" description="Simulation results showing successful passenger interaction under various conditions" type="SupportedBy" />
                </goal>
            </strategy>
        </goal>
    </strategy>
</goal>
```

## A.5 THE U-SHIFT CONTROL SYSTEM'S SAFETY ANALYSIS AND SYSTEM DESCRIPTION

Listing A.5: Safety Analysis and system description for the U-SHIFT control system

```

## System Description and Safety Analysis: U-SHIFT Control System

---

#### 1. System Definition

**System Overview**:
The Control System is the central decision-making unit of the U-SHIFT vehicle, responsible for processing sensor data, managing interactions with other subsystems (Drive, Onboarding, Communication, and Emergency systems), and issuing real-time commands. It ensures coordinated operations for vehicle navigation, passenger safety, and emergency handling, playing a critical role in vehicle automation and system safety.

**System Boundaries**:
- **Subsystems**:
  - **Data Processing Subsystem**: Collects and processes data from sensors and subsystem interfaces.
  - **Decision-Making Subsystem**: Analyzes data to make real-time operational decisions.
  - **Command Interface Subsystem**: Issues commands to Drive, Onboarding, and other connected systems.
  - **Diagnostics Subsystem**: Continuously monitors system performance and detects faults.

- **Components**:
```

```
  - **Central Processor**: High-performance processor for handling complex data processing and decision-making.
  - **Communication Module**: Interface for data exchange between subsystems and external operators.
  - **Diagnostics Interface**: Monitors system health and provides alerts for detected faults.
  - **Power Management Unit**: Ensures continuous power supply and manages power distribution across control components.

---

#### 2. Operational Environment

**Physical Environment**:
The Control System operates in urban and suburban areas, handling complex road interactions, traffic scenarios, and passenger onboarding and offboarding at bus stops and other designated locations. It must adapt to:
- **Road Types**: Asphalt, concrete, and varied urban road structures.
- **Weather Conditions**: Reliable operation in all-weather conditions, including rain, fog, and moderate snow.
- **Lighting Conditions**: Effective operation in both day and night settings.

**Operational Scenarios**:
- **Scenario 1**: Decision-making for vehicle stopping at a bus stop while ensuring safe boarding.
- **Scenario 2**: Real-time hazard detection and response when obstacles are detected on the path.
- **Scenario 3**: Coordination with Drive System for speed adjustments during passenger onboarding.

**Human Interaction**:
The Control System interacts with operators and passengers through:
- **Operator Alerts**: Provides real-time updates on system status and emergency alerts.
- **Passenger Notifications**: Issues notifications when there are sudden changes in speed or route deviations for safety.

---

#### 3. Functional and Performance Requirements

**Functional Requirements**:
- **Data Integration**: Gather and process data from all vehicle subsystems and sensors.
- **Decision-Making**: Analyze real-time data to identify hazards, manage vehicle speed, and control passenger access.
- **Command Execution**: Issue timely and accurate commands to the Drive, Onboarding, Communication, and Emergency systems.

```

```
**Performance Requirements**:
- **Response Time**: Must process and respond to critical inputs within 100ms.
- **Data Accuracy**: 99% accuracy in data interpretation from subsystems.
- **System Uptime**: 99.9% availability in normal operational conditions.

---

#### 4. Safety-Critical Functions

**Identification**:
- **Function 1**: Real-time hazard detection and response.
- **Function 2**: Safe coordination of vehicle movement and passenger access.
- **Function 3**: Emergency command execution, including stopping and notifying external systems.

**Failure Modes**:
- **Mode 1**: Loss of data processing capability.
- **Mode 2**: Command interface malfunction, leading to loss of control over subsystems.
- **Mode 3**: Diagnostics subsystem failure, resulting in undetected faults.

---

#### 5. System Architecture and Design

**Architecture Overview**:
The Control System is designed with modular components to ensure efficient and fault-tolerant data processing, decision-making, and command issuance. It connects seamlessly with other subsystems to maintain smooth and safe operations across all driving scenarios.

**Design Principles**:
- **Redundancy**: Redundant data channels for critical commands to Drive and Emergency systems to prevent single points of failure.
- **Fail-safe Operation**: Defaults to a safe braking and halt mode if critical processing faults are detected.

**Safety Mechanisms**:
- **Real-time Diagnostics**: Constantly monitors system health and performance.
- **Multi-level Decision Checks**: Cross-verifies sensor and subsystem data before executing high-impact commands.

---

#### 6. Implementation Details

**Software**:
The Control System software is divided into modules for:
```

```
1. **Data Fusion Module**: Integrates multi-source sensor data for comprehensive situational awareness.
2. **Decision-Making Algorithm**: Analyzes data and determines safe actions based on pre-defined safety rules.
3. **Command Dispatcher**: Issues commands to Drive, Onboarding, and Emergency systems in real-time.

**Hardware**:
- **Central Processor**: A high-performance unit (e.g., NVIDIA Jetson or similar) dedicated to processing large volumes of data and executing decisions in milliseconds.
- **Communication Module**: Allows high-speed data transfer with other subsystems and external control systems.
- **Diagnostics Interface**: Continuously checks for system errors and provides fault detection.

**Integration and Configuration Management**:
The Control System integrates with the U-SHIFT vehicles central control infrastructure, enabling seamless coordination with all other subsystems. Configuration control includes routine software updates, strict version management, and secure boot processes.

---

#### 7. Verification and Validation

**Testing Procedures**:
- **Unit Testing**: Verifies individual functions such as data processing and decision-making logic.
- **Integration Testing**: Assesses full system operation, focusing on control coordination during navigation and emergency events.
- **Stress Testing**: Evaluates system performance under heavy data load and in extreme conditions.

**Validation Methods**:
- **Simulation Testing**: Tests system response using simulated driving and hazard scenarios.
- **Field Testing**: Real-world testing across urban routes, focusing on hazard detection and passenger safety coordination.

**Test Results**:
- Hazard detection accuracy: 97% within specified response times.
- Command execution latency: <100ms in 98% of test cases.

---

#### 8. Maintenance and Operation

**Maintenance Procedures**:
```

```
- **Routine Diagnostics**: Regular checks on processing and communication modules to ensure reliability.
- **Software Updates**: Scheduled updates to improve performance and add new features.
- **Operator Training**: Provides training on manual overrides and system diagnostics.

**Operational Procedures**:
- The Control System operates autonomously but can be remotely managed or overridden by authorized operators.
- The system logs all actions and events for later analysis and diagnostics.

**Incident Response**:
- Any emergency stop or loss of data generates an automatic report for immediate review by maintenance personnel.

---

#### 9. Regulatory and Compliance Information

**Applicable Standards**:
- **ISO 26262**: Functional safety standards for road vehicles.
- **UNECE R79**: Standards for vehicle steering and control systems.
- **SAE J3016**: Defines automation levels, with U-SHIFT Control System aligning with levels 3-4.

**Compliance Strategy**:
- The Control System undergoes regular compliance testing to meet standards, with periodic audits and assessments.

---

#### 10. Safety Analysis

**Purpose**:
The purpose of this safety analysis is to identify potential hazards in the Control System, assess associated risks, and define mitigations to ensure safe and reliable operations.

---

#### 10.1 Hazard Identification

| **ID**   | **Hazard**                          | **Description**                                   |
|---------|-------------------------------------|-------------------------------------------------------------------------
| **H1**   | Data processing failure             | Loss of data processing leading to delayed or inaccurate decisions. |
| **H2**   | Command interface loss              | Inability to send commands to Drive or Emergency systems.        |
```

```
| **H3**  | Fault in diagnostics              | Failure to detect critical
    ↪ faults, increasing risk of undetected issues. |
| **H4**  | Sensor data misinterpretation     | Incorrect data processing
    ↪ could lead to unsafe commands.          |
| **H5**  | Loss of communication with subsystems | Disconnection from Drive,
    ↪ Onboarding, or Emergency systems.       |

---

#### 10.2 Risk Assessment

| **Hazard ID** | **Likelihood** | **Severity** | **Risk Rating** |
    ↪ **Mitigation Strategy**                                |
|---------------|----------------|--------------|-----------------|-----------------------------
| **H1**        | Likely (3)     | Severe (4)   | High (12)       | Redundant
    ↪ data channels and fallback processing units.           |
| **H2**        | Occasional (2) | Critical (5) | High (10)       | Backup
    ↪ interface for critical commands.                          |
| **H3**        | Rare (1)       | Severe (4)   | Medium (4)      | Enhanced
    ↪ diagnostics with regular system checks.               |
| **H4**        | Occasional (2) | Major (3)    | Medium (6)      |
    ↪ Cross-validation of sensor data across multiple sources.      |
| **H5**        | Unlikely (1)   | Major (3)    | Low (3)         | Heartbeat
    ↪ signals to monitor subsystem connectivity.             |

---

#### 10.3 Risk Mitigation Strategies

- **H1 - Data Processing Failure**: Implement redundant processors and
    ↪ error-checking algorithms.
- **H2 - Command Interface Loss**: Configure backup command channels for
    ↪ essential functions.
- **H3 - Diagnostics Fault**: Enhance diagnostics with multi-level checks and
    ↪ real-time fault reporting.
- **H4 - Sensor Data Misinterpretation**: Apply data fusion techniques for
    ↪ improved accuracy in decision-making.
- **H5 - Loss of Communication with Subsystems**: Utilize heartbeat checks
    ↪ and redundancy in data communication links.
```

## A.6 THE U-SHIFT CONTROL SYSTEM'S SAFETY CASES IN XML BASED ON GSN

Listing A.6: Safety Cases for the U-SHIFT Control system

```
<goal id="G1" description="The U-SHIFT Control System is acceptably safe"
    ↪ type="SupportedBy">
    <context id="C1" description="System Overview: The Control System is the
        ↪ central decision-making unit of the U-SHIFT vehicle, responsible
        ↪ for processing sensor data, managing interactions with other
```

```
    ↪ subsystems (Drive, Onboarding, Communication, and Emergency
    ↪ systems), and issuing real-time commands" type="InContextOf"/>
<context id="C2" description="System Boundaries: The Control System
    ↪ operates within the U-SHIFT vehicle, interacting with various
    ↪ subsystems to ensure safe and efficient operations"
    ↪ type="InContextOf"/>
<strategy id="S1" description="Argument by decomposition of Control
    ↪ System functionality into three core modules: Data Processing,
    ↪ Decision-Making, and Command Interface" type="SupportedBy">
    <goal id="G2" description="Correct and complete implementation of the
        ↪ Data Processing Module" type="SupportedBy">
        <context id="C3" description="Data Processing Module: Responsible
            ↪ for collecting and processing data from sensors and
            ↪ subsystem interfaces" type="InContextOf"/>
        <assumption id="A1" description="All sensors and subsystem
            ↪ interfaces provide accurate and reliable data under
            ↪ nominal and off-nominal conditions" type="InContextOf"/>
        <justification id="J1" description="The Data Processing Module
            ↪ must accurately interpret data from various sources to
            ↪ ensure safe and efficient system operation"
            ↪ type="InContextOf"/>
        <solution id="E1" description="Implementation of data validation
            ↪ and correction algorithms to ensure data integrity"
            ↪ type="SupportedBy"/>
    </goal>
    <goal id="G3" description="Correct and complete implementation of the
        ↪ Decision-Making Module" type="SupportedBy">
        <context id="C4" description="Decision-Making Module: Responsible
            ↪ for analyzing data to make real-time operational
            ↪ decisions" type="InContextOf"/>
        <assumption id="A2" description="All decision-making algorithms
            ↪ are designed to account for uncertainties and variations
            ↪ in system dynamics" type="InContextOf"/>
        <justification id="J2" description="The Decision-Making Module
            ↪ must accurately analyze data to make safe and effective
            ↪ decisions under all operational conditions"
            ↪ type="InContextOf"/>
        <solution id="E2" description="Implementation of decision-making
            ↪ algorithms with built-in fault tolerance and redundancy"
            ↪ type="SupportedBy"/>
    </goal>
    <goal id="G4" description="Correct and complete implementation of the
        ↪ Command Interface Module" type="SupportedBy">
        <context id="C5" description="Command Interface Module:
            ↪ Responsible for issuing commands to Drive, Onboarding,
            ↪ Communication, and Emergency systems" type="InContextOf"/>
        <assumption id="A3" description="All commands are correctly
            ↪ issued and executed under all operational conditions"
            ↪ type="InContextOf"/>
```

```
            <justification id="J3" description="The Command Interface Module
                ↪ must ensure that all commands are correctly issued and
                ↪ executed to maintain system safety" type="InContextOf"/>
            <solution id="E3" description="Implementation of command
                ↪ validation and execution algorithms to ensure correctness
                ↪ and reliability" type="SupportedBy"/>
        </goal>
    </strategy>
    <strategy id="S2" description="Argument by hazard mitigation"
        ↪ type="SupportedBy">
        <goal id="G5" description="Hazard H1: Data processing failure is
            ↪ mitigated" type="SupportedBy">
            <context id="C6" description="Hazard H1: Loss of data processing
                ↪ leading to delayed or inaccurate decisions"
                ↪ type="InContextOf"/>
            <assumption id="A4" description="The Data Processing Module is
                ↪ designed to recover from data processing failures"
                ↪ type="InContextOf"/>
            <justification id="J4" description="The Data Processing Module
                ↪ must recover from data processing failures to ensure
                ↪ system safety" type="InContextOf"/>
            <solution id="E4" description="Implementation of data processing
                ↪ recovery algorithms" type="SupportedBy"/>
        </goal>
        <goal id="G6" description="Hazard H2: Command interface loss is
            ↪ mitigated" type="SupportedBy">
            <context id="C7" description="Hazard H2: Inability to send
                ↪ commands to Drive or Emergency systems"
                ↪ type="InContextOf"/>
            <assumption id="A5" description="The Command Interface Module is
                ↪ designed to recover from command interface losses"
                ↪ type="InContextOf"/>
            <justification id="J5" description="The Command Interface Module
                ↪ must recover from command interface losses to ensure
                ↪ system safety" type="InContextOf"/>
            <solution id="E5" description="Implementation of command
                ↪ interface recovery algorithms" type="SupportedBy"/>
        </goal>
        <goal id="G7" description="Hazard H3: Fault in diagnostics is
            ↪ mitigated" type="SupportedBy">
            <context id="C8" description="Hazard H3: Failure to detect
                ↪ critical faults" type="InContextOf"/>
            <assumption id="A6" description="The Diagnostics Module is
                ↪ designed to detect critical faults" type="InContextOf"/>
            <justification id="J6" description="The Diagnostics Module must
                ↪ detect critical faults to ensure system safety"
                ↪ type="InContextOf"/>
            <solution id="E6" description="Implementation of diagnostics
                ↪ algorithms" type="SupportedBy"/>
        </goal>
```

```
        <goal id="G8" description="Hazard H4: Sensor data misinterpretation
            ↪ is mitigated" type="SupportedBy">
            <context id="C9" description="Hazard H4: Incorrect data
                ↪ processing could lead to unsafe commands"
                ↪ type="InContextOf"/>
            <assumption id="A7" description="The Data Processing Module is
                ↪ designed to correct sensor data misinterpretation"
                ↪ type="InContextOf"/>
            <justification id="J7" description="The Data Processing Module
                ↪ must correct sensor data misinterpretation to ensure
                ↪ system safety" type="InContextOf"/>
            <solution id="E7" description="Implementation of sensor data
                ↪ correction algorithms" type="SupportedBy"/>
        </goal>
        <goal id="G9" description="Hazard H5: Loss of communication with
            ↪ subsystems is mitigated" type="SupportedBy">
            <context id="C10" description="Hazard H5: Disconnection from
                ↪ Drive, Onboarding, or Emergency systems"
                ↪ type="InContextOf"/>
            <assumption id="A8" description="The Communication Module is
                ↪ designed to recover from communication losses"
                ↪ type="InContextOf"/>
            <justification id="J8" description="The Communication Module must
                ↪ recover from communication losses to ensure system safety"
                ↪ type="InContextOf"/>
            <solution id="E8" description="Implementation of communication
                ↪ recovery algorithms" type="SupportedBy"/>
        </goal>
    </strategy>
</goal>
```

## A.7 THE U-SHIFT COMMUNICATION SYSTEM'S SAFETY ANALYSIS AND SYSTEM DESCRIPTION

Listing A.7: Safety Analysis and System description for the U-SHIFT Communication system

```

## System Description and Safety Analysis: U-SHIFT Communication System

---

#### 1. System Definition

**System Overview**:
The Communication System in the U-SHIFT vehicle enables information exchange
    ↪ between the vehicle, external operators, and passengers. It handles
    ↪ data related to operational status, emergency alerts, and passenger
    ↪ notifications. This system is crucial for ensuring the vehicle
```

```
operates safely and transparently in various driving conditions, facilitating remote control and monitoring when required.

**System Boundaries**:
- **Subsystems**:
  - **Operator Interface Subsystem**: Connects with external operators for real-time updates and remote commands.
  - **Passenger Notification Subsystem**: Provides safety and status updates to passengers inside the capsule.
  - **Data Encryption Subsystem**: Secures data transfer between the vehicle and external systems.
  - **System Monitoring Subsystem**: Tracks communication health and detects disruptions.

- **Components**:
  - **Wireless Communication Module**: Supports cellular, Wi-Fi, and V2X (Vehicle-to-Everything) connectivity for real-time data exchange.
  - **Operator Control Interface**: Provides operators with real-time access to vehicle status and control features.
  - **Passenger Display Unit**: Displays information to passengers, including route details, stops, and emergency notifications.
  - **Encryption Module**: Protects data transmitted between the vehicle and external entities.

---

#### 2. Operational Environment

**Physical Environment**:
The Communication System operates in urban, suburban, and intercity environments, handling a range of external interactions and data transmissions:
- **Network Conditions**: Must function reliably across cellular, Wi-Fi, and low-latency V2X networks.
- **Weather and Signal Interference**: Capable of maintaining signal integrity in adverse weather and areas with high signal interference.
- **Operational Range**: Supports seamless connectivity over long distances in urban and rural areas.

**Operational Scenarios**:
- **Scenario 1**: Communicating vehicle status updates and alerts to a remote operator.
- **Scenario 2**: Issuing route and safety information to passengers during transit.
- **Scenario 3**: Encrypted communication with external traffic systems for coordinated movements and alerts.

**Human Interaction**:
Operators and passengers receive relevant information through the Communication System:
```

```
- **Operator Alerts**: Provides real-time status updates and alerts for intervention when necessary.
- **Passenger Notifications**: Displays safety messages, route updates, and emergency instructions.

---

#### 3. Functional and Performance Requirements

**Functional Requirements**:
- **Reliable Data Exchange**: Maintain continuous data communication with external operators and subsystems.
- **Passenger Information Delivery**: Notify passengers of route information, stops, and emergency situations.
- **Data Security**: Ensure all communication with external systems is encrypted and protected against unauthorized access.

**Performance Requirements**:
- **Latency**: Maintain a maximum latency of 100ms for critical operator commands.
- **Data Integrity**: Ensure 99.9% data integrity in real-time communications.
- **Uptime**: Achieve 99% communication uptime under normal conditions.

---

#### 4. Safety-Critical Functions

**Identification**:
- **Function 1**: Secure transmission of vehicle data to operators.
- **Function 2**: Real-time notification of passengers for safety and operational updates.
- **Function 3**: Reliable encryption of all transmitted data to protect from unauthorized access.

**Failure Modes**:
- **Mode 1**: Loss of network connectivity, resulting in inability to send/receive data.
- **Mode 2**: Delay in data transmission, leading to delayed notifications or commands.
- **Mode 3**: Unauthorized access to the communication channel, risking data security.

---

#### 5. System Architecture and Design

**Architecture Overview**:
The Communication System is structured with modular components to enable seamless data transfer between the U-SHIFT vehicle, passengers, and
```

```
external control centers. It includes secure data pathways, ensuring both reliability and data protection.

**Design Principles**:
- **Redundancy**: Dual-channel communication modules to prevent single points of failure.
- **Fail-safe Operation**: In case of network failure, default to notifying operators via backup communication.

**Safety Mechanisms**:
- **Data Encryption**: All communication with external systems is encrypted to protect sensitive information.
- **Connectivity Check**: Periodic checks on network status to ensure continuous data flow.

---

#### 6. Implementation Details

**Software**:
The Communication System software includes:
1. **Data Encryption Module**: Encrypts data for secure external transmission.
2. **Real-Time Monitoring Module**: Tracks network quality and alerts operators of any disruption.
3. **Notification System**: Manages passenger information display for route and safety updates.

**Hardware**:
- **Communication Interface**: Multi-mode wireless modules supporting cellular, Wi-Fi, and V2X.
- **Processor**: Manages data encryption and real-time monitoring of communication status.
- **Display Interface**: Screens for displaying passenger notifications and alerts.

**Integration and Configuration Management**:
Integrated with the v e h i c l e s control and emergency systems, the Communication System undergoes configuration management to ensure consistency in updates and secure software version control.

---

#### 7. Verification and Validation

**Testing Procedures**:
- **Unit Testing**: Tests each module individually, including encryption and notification systems.
- **Integration Testing**: Verifies the full communication flow from vehicle to operator and passenger notifications.
```

```
- **Stress Testing**: Evaluates communication resilience under high traffic, poor connectivity, and interference.

**Validation Methods**:
- **Simulation Testing**: Tests data transmission in simulated urban and rural environments to evaluate connectivity.
- **Field Testing**: Real-world testing with focus on maintaining data integrity and low latency.

**Test Results**:
- Connectivity uptime: 99% in urban and rural tests.
- Data transmission latency: <100ms in 95% of test cases.

---

#### 8. Maintenance and Operation

**Maintenance Procedures**:
- **Routine Diagnostics**: Regular checks of communication modules and encryption protocols.
- **Firmware Updates**: Scheduled updates to enhance performance and add new features.
- **Security Audits**: Periodic audits to verify compliance with data security standards.

**Operational Procedures**:
- The Communication System operates autonomously, with remote management capability by operators.
- All transmitted data is logged for diagnostic and maintenance purposes.

**Incident Response**:
- Any communication loss or security breach generates an immediate alert to operators and logs a report for review.

---

#### 9. Regulatory and Compliance Information

**Applicable Standards**:
- **ISO 26262**: Functional safety standard for road vehicles.
- **UNECE R10**: Electromagnetic compatibility standards for in-vehicle communication.
- **SAE J3061**: Cybersecurity guidelines for vehicle communication.

**Compliance Strategy**:
- The Communication System undergoes regular testing to ensure compliance with data security and safety standards. Independent audits verify adherence to industry standards.

---
```

```

#### 10. Safety Analysis

**Purpose**:
Identify potential hazards in the Communication System, assess associated risks, and define mitigations to ensure reliable and secure data transfer.

---

#### 10.1 Hazard Identification

| **ID**  | **Hazard**                           | **Description**                                                |
|---------|--------------------------------------|--------------------------------------------------------------
| **H1**  | Network connectivity loss            | Loss of communication with operators, risking vehicle safety.    |
| **H2**  | Data transmission delay              | Delay in sending/receiving data could lead to safety risks.      |
| **H3**  | Unauthorized access                  | Data breach could expose vehicle controls to external threats.   |
| **H4**  | Data corruption                      | Inaccurate data transmission affecting passenger notifications.  |
| **H5**  | Incomplete passenger notifications   | Failure to inform passengers of critical safety information.     |

---

#### 10.2 Risk Assessment

| **Hazard ID** | **Likelihood** | **Severity** | **Risk Rating** | **Mitigation Strategy**                                   |
|---------------|----------------|--------------|-----------------|-----------------------------------------------------------
| **H1**        | Likely (3)     | Severe (4)   | High (12)       | Redundant communication channels and real-time monitoring.    |
| **H2**        | Occasional (2) | Critical (5) | High (10)       | Prioritize critical data and optimize transmission protocols. |
| **H3**        | Rare (1)       | Critical (5) | Medium (5)      | Use multi-layer encryption and secure access protocols.       |
| **H4**        | Occasional (2) | Major (3)    | Medium (6)      | Implement error-checking and correction algorithms.           |
| **H5**        | Unlikely (1)   | Moderate (3) | Low (3)         | Periodic checks on notification system integrity.             |

---

#### 10.3 Risk Mitigation Strategies

- **H1 - Network Connectivity Loss**: Configure dual-channel connectivity with automatic failover mechanisms.
```

```
- **H2 - Data Transmission Delay**: Implement priority-based data
    ↪ transmission for time-sensitive information.
- **H3 - Unauthorized Access**: Apply end-to-end encryption and access
    ↪ control mechanisms.
- **H4 - Data Corruption**: Integrate error-checking and validation
    ↪ techniques for data integrity.
- **H5 - Incomplete Passenger Notifications**: Regularly test notification
    ↪ systems to ensure reliable operation and messaging.
```

## A.8 THE U-SHIFT COMMUNICATION SYSTEM'S SAFETY CASES IN XML BASED ON GSN

Listing A.8: Safety Cases for the U-SHIFT Communication system

```
<goal id="G1" description="The Communication System in the U-SHIFT vehicle
    ↪ ensures reliable and secure data transfer between the vehicle,
    ↪ external operators, and passengers" type="SupportedBy">
    <context id="C1" description="System Overview: The Communication System
        ↪ in the U-SHIFT vehicle enables information exchange between the
        ↪ vehicle, external operators, and passengers" type="InContextOf"/>
    <context id="C2" description="System Boundaries: Includes Operator
        ↪ Interface Subsystem, Passenger Notification Subsystem, Data
        ↪ Encryption Subsystem, and System Monitoring Subsystem"
        ↪ type="InContextOf"/>
    <assumption id="A1" description="The system-level requirements for the
        ↪ Communication System have been validated through stakeholder
        ↪ review and are consistent with the overall mission objectives"
        ↪ type="InContextOf"/>
    <justification id="J1" description="The purpose of this safety case is to
        ↪ demonstrate that the Communication System design and
        ↪ implementation meet all safety requirements, ensuring reliable and
        ↪ secure data transfer under all operational conditions"
        ↪ type="InContextOf"/>

    <strategy id="S1" description="Argument by decomposition of Communication
        ↪ System functionality into three core modules: Operator Interface,
        ↪ Passenger Notification, and Data Encryption" type="SupportedBy">
        <goal id="G2" description="Correct and complete implementation of the
            ↪ Operator Interface Module, ensuring reliable communication
            ↪ with external operators" type="SupportedBy">
            <context id="C3" description="Operator Interface Module is
                ↪ responsible for connecting with external operators for
                ↪ real-time updates and remote commands" type="InContextOf"/>
            <assumption id="A2" description="The Operator Interface Module
                ↪ design and implementation are based on established
                ↪ communication protocols and standards" type="InContextOf"/>
            <justification id="J2" description="Argument over the design and
                ↪ implementation of the Operator Interface Module,
```

```
                ↪ validating its functionality and performance under all
                ↪ operational scenarios" type="InContextOf"/>

            <strategy id="S2" description="Argument by functional validation
                ↪ of Operator Interface Module algorithms, including data
                ↪ encryption and secure access control" type="SupportedBy">
                <goal id="G3" description="Operator Interface Module
                    ↪ algorithms are validated through simulation and formal
                    ↪ analysis, ensuring reliable and secure communication
                    ↪ with external operators" type="SupportedBy">
                    <context id="C4" description="Simulation results and
                        ↪ formal analysis reports are available to validate
                        ↪ the performance and security of the Operator
                        ↪ Interface Module algorithms" type="InContextOf"/>
                    <assumption id="A3" description="The simulation
                        ↪ environment accurately models the operational
                        ↪ conditions expected for the Operator Interface
                        ↪ Module" type="InContextOf"/>
                    <justification id="J3" description="The validation of the
                        ↪ Operator Interface Module algorithms is critical
                        ↪ to ensuring reliable and secure communication with
                        ↪ external operators" type="InContextOf"/>

                    <solution id="E1" description="Simulation results and
                        ↪ formal analysis reports for the Operator Interface
                        ↪ Module algorithms, validating their functionality
                        ↪ and security under all operational conditions"
                        ↪ type="SupportedBy"/>
                </goal>
            </strategy>
        </goal>

        <goal id="G4" description="Correct and complete implementation of the
            ↪ Passenger Notification Module, ensuring reliable and secure
            ↪ communication with passengers" type="SupportedBy">
            <context id="C5" description="Passenger Notification Module is
                ↪ responsible for notifying passengers of route information,
                ↪ stops, and emergency situations" type="InContextOf"/>
            <assumption id="A4" description="The Passenger Notification
                ↪ Module design and implementation are based on established
                ↪ communication protocols and standards" type="InContextOf"/>
            <justification id="J4" description="Argument over the design and
                ↪ implementation of the Passenger Notification Module,
                ↪ validating its functionality and performance under all
                ↪ operational scenarios" type="InContextOf"/>

            <strategy id="S3" description="Argument by functional validation
                ↪ of Passenger Notification Module algorithms, including
                ↪ data encryption and secure access control"
                ↪ type="SupportedBy">
```

```
            <goal id="G5" description="Passenger Notification Module
                ↪ algorithms are validated through simulation and formal
                ↪ analysis, ensuring reliable and secure communication
                ↪ with passengers" type="SupportedBy">
                <context id="C6" description="Simulation results and
                    ↪ formal analysis reports are available to validate
                    ↪ the performance and security of the Passenger
                    ↪ Notification Module algorithms"
                    ↪ type="InContextOf"/>
                <assumption id="A5" description="The simulation
                    ↪ environment accurately models the operational
                    ↪ conditions expected for the Passenger Notification
                    ↪ Module" type="InContextOf"/>
                <justification id="J5" description="The validation of the
                    ↪ Passenger Notification Module algorithms is
                    ↪ critical to ensuring reliable and secure
                    ↪ communication with passengers" type="InContextOf"/>

                <solution id="E2" description="Simulation results and
                    ↪ formal analysis reports for the Passenger
                    ↪ Notification Module algorithms, validating their
                    ↪ functionality and security under all operational
                    ↪ conditions" type="SupportedBy"/>
            </goal>
        </strategy>
    </goal>

    <goal id="G6" description="Correct and complete implementation of the
        ↪ Data Encryption Module, ensuring secure communication with
        ↪ external systems" type="SupportedBy">
        <context id="C7" description="Data Encryption Module is
            ↪ responsible for encrypting data for secure external
            ↪ transmission" type="InContextOf"/>
        <assumption id="A6" description="The Data Encryption Module
            ↪ design and implementation are based on established
            ↪ cryptographic protocols and standards" type="InContextOf"/>
        <justification id="J6" description="Argument over the design and
            ↪ implementation of the Data Encryption Module, validating
            ↪ its functionality and performance under all operational
            ↪ scenarios" type="InContextOf"/>

        <strategy id="S4" description="Argument by functional validation
            ↪ of Data Encryption Module algorithms, including key
            ↪ management and secure data transmission"
            ↪ type="SupportedBy">
            <goal id="G7" description="Data Encryption Module algorithms
                ↪ are validated through simulation and formal analysis,
                ↪ ensuring secure communication with external systems"
                ↪ type="SupportedBy">
```

```
                    <context id="C8" description="Simulation results and formal analysis reports are available to validate the performance and security of the Data Encryption Module algorithms" type="InContextOf"/>
                    <assumption id="A7" description="The simulation environment accurately models the operational conditions expected for the Data Encryption Module" type="InContextOf"/>
                    <justification id="J7" description="The validation of the Data Encryption Module algorithms is critical to ensuring secure communication with external systems" type="InContextOf"/>

                    <solution id="E3" description="Simulation results and formal analysis reports for the Data Encryption Module algorithms, validating their functionality and security under all operational conditions" type="SupportedBy"/>
                </goal>
            </strategy>
        </goal>
    </strategy>
</goal>
```

## A.9 API CODE FOR RUNNING LLAMA-3.1-405B-INSTRUCT FROM NVIDIA NIM FOR GENERATION OF SAFETY ANALYSES AND SYSTEM DESIGNS

Listing A.9: API Code for Running Llama-3.1-405b-Instruct from Nvidia NIM

```python

import requests
from openai import OpenAI

client = OpenAI(
    base_url = "https://integrate.api.nvidia.com/v1",
    api_key = "$API_KEY_REQUIRED_IF_EXECUTING_OUTSIDE_NGC"
)

# Function to interact with OpenAI API
def generate_safety_analysis(system_name):
    completion1 = client.chat.completions.create(
        model="meta/llama-3.1-405b-instruct",
        messages=[
            {"role":"system","content":System_Prompts_new.SYSTEM_PROMPT_Saf_Anal_Sys_Desc},
            {"role":"user","content":"Please generate a high-level safety analyis and system description for the following system: "+ system_name}
        ],
        temperature=0.2,
```

```
        top_p=0.7,
        max_tokens=16384,
        stream=True
    )

    output_chunks_1 = []  # Clear the list before each generation
    for chunk in completion1:
        if chunk.choices[0].delta.content is not None:
            output_chunks_1.append(chunk.choices[0].delta.content)

    saf_anal_sys_desc = ''.join(output_chunks_1)

    return saf_anal_sys_desc
```

## A.10 API CODE FOR RUNNING LLAMA-3.1-405B-INSTRUCT FROM NVIDIA NIM FOR GENERATION OF SAFETY CASES IN XML

Listing A.10: API Code for Running Llama-3.1-405b-Instruct for Safety Cases in XML generation

```

import requests
from openai import OpenAI

client = OpenAI(
    base_url = "https://integrate.api.nvidia.com/v1",
    api_key = "$API_KEY_REQUIRED_IF_EXECUTING_OUTSIDE_NGC"
)

# Function to interact with OpenAI API
def generate_safety_cases_xml(safety_analysis):
    completion2 = client.chat.completions.create(
        model="meta/llama-3.1-405b-instruct",
        messages=[
            {"role":"system","content":System_Prompts_new.SYSTEM_PROMPT_SAFETY_CASES},
            {"role":"user","content": safety_analysis}
        ],
        temperature=0.2,
        top_p=0.7,
        max_tokens=16384,
        stream=True
    )

    output_chunks_2 = [] # Clear the list before each generation
    for chunk in completion2:
        if chunk.choices[0].delta.content is not None:
            output_chunks_2.append(chunk.choices[0].delta.content)

    safety_cases = ''.join(output_chunks_2)
```

```

    return safety_cases
```

## A.11 CODE SOURCE FOR FINE-TUNING LLAMA-3-8B

```
#### Step 1: Install the Required Packages

!pip install unsloth
# Also get the latest nightly Unsloth!
!pip uninstall unsloth -y && pip install --upgrade --no-cache-dir
    ↪ "unsloth[colab-new] @ git+https://github.com/unslothai/unsloth.git"


from unsloth import FastLanguageModel
import torch
from datasets import load_dataset
from trl import SFTTrainer
from transformers import TrainingArguments
from peft import AutoPeftModelForCausalLM
from transformers import AutoTokenizer


#### Step 2: Load the Pre-trained model and tokenizer
model_name = "unsloth/Meta-Llama-3.1-8B-bnb-4bit"
max_seq_length = 65536
dtype =  'bfloat16'
load_in_4bit = True

model, tokenizer = FastLanguageModel.from_pretrained(
    model_name=model_name,
    max_seq_length=max_seq_length,
    dtype=torch.bfloat16,
    load_in_4bit=load_in_4bit,
)


#### Step 3: Define a prompt Format and load Format Dataset

alpaca_prompt = """### Safety analysis:
{}

#### Safety Cases in XML:
{}"""

EOS_TOKEN = tokenizer.eos_token # Must add EOS_TOKEN

# Format the dataset
def format_data(example):
```

```
    Safety_analyses       = example["Safety Analysis"]
    Safety_cases_in_XMLs     = example["Safety Cases in XML"]
    texts = []
    for safety_analysis, Safety_cases_in_XML in zip(Safety_analyses, Safety_cases_in_XMLs):
        # Must add EOS_TOKEN, otherwise your generation will go on forever!
        text = alpaca_prompt.format(safety_analysis, Safety_cases_in_XML) + EOS_TOKEN
        texts.append(text)
    return { "text" : texts, }
pass

# Load the dataset
dataset = load_dataset("Max491/New_Version_XML_Safety_Cases_Archive", split = "train")

dataset = dataset.map(format_data, remove_columns=['System Names'], batched = True,)



#### Step 4: Apply LoRA Fine-Tuning

model = FastLanguageModel.get_peft_model(
    model,
    r = 16,
    target_modules = ["q_proj", "k_proj", "v_proj", "o_proj",
                      "gate_proj", "up_proj", "down_proj",],
    lora_alpha = 16,
    bias = "none",
    use_gradient_checkpointing = "unsloth",
    random_state = 3407,
    use_rslora = False,
    loftq_config = None,
)



#### Step 5: Configure Training Settings

trainer = SFTTrainer(
    model = model,
    tokenizer = tokenizer,
    train_dataset = dataset,
    dataset_text_field = "text",
    max_seq_length = max_seq_length,
    dataset_num_proc = 2,
    packing = False,
    args = TrainingArguments(
        per_device_train_batch_size = 2,
```

```
        gradient_accumulation_steps = 4,
        warmup_steps = 5,
        max_steps = 60,
        learning_rate = 2e-4,
        fp16 = not torch.cuda.is_bf16_supported(),
        bf16 = torch.cuda.is_bf16_supported(),
        logging_steps = 1,
        optim = "adamw_8bit",
        weight_decay = 0.01,
        lr_scheduler_type = "linear",
        seed = 3407,
        output_dir = "outputs",
    ),
)

#### Step 6: Show current memory stats

gpu_stats = torch.cuda.get_device_properties(0)
start_gpu_memory = round(torch.cuda.max_memory_reserved() / 1024 / 1024 /
    ↪ 1024, 3)
max_memory = round(gpu_stats.total_memory / 1024 / 1024 / 1024, 3)
print(f"GPU = {gpu_stats.name}. Max memory = {max_memory} GB.")
print(f"{start_gpu_memory} GB of memory reserved.")


#### Step 7: Train the Model
trainer_stats = trainer.train()
```

## A.12 CODE-SOURCE FOR SAVING AND PUSHING THE LORA AND TOKENIZER ON HUGGING FACE HUB

```


#### Step 1: Download the LoRA and the tokenizer locally

model.save_pretrained("save/model")
tokenizer.save_pretrained("save/tokenizer")
model.config.save_pretrained("save/model")

#### Step 2: Push the LoRA and the tokenizer on Hugging Face Hub

# Name repository on Hugging Face
model_name = "Max491/Safety_Cases_Generator_new"

# Log in to Hugging Face
huggingface_token = "hf_....."  #Insert HF token
login(huggingface_token)

```

## A.12 CODE-SOURCE FOR SAVING AND PUSHING THE LORA AND TOKENIZER ON HUGGING FACE HUB

```
# Push the LoRA and Tokenizer
model.push_to_hub(model_name)
tokenizer.push_to_hub(model_name)
```