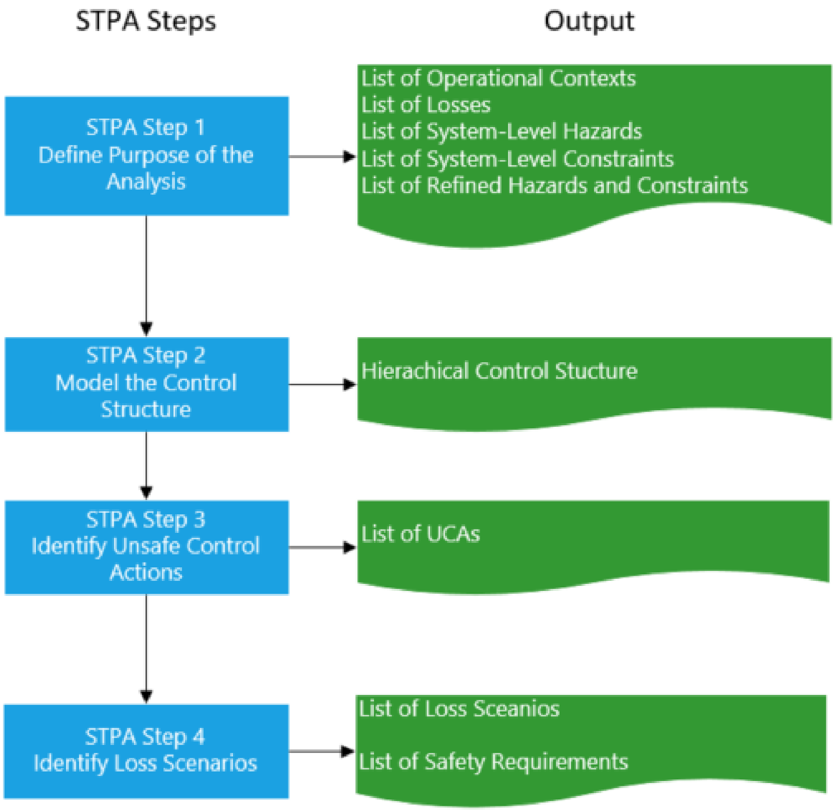
RESEARCH REPORT

VTT-R-00848-23


STPA Steps
Output
STPA Step 1
Define Purpose of the Analysis
List of Operational Contexts
List of Losses
List of System-Level Hazards
List of System-Level Constraints
List of Refined Hazards and Constraints
STPA Step 2
Model the Control Structure
Hierachical Control Stucture
STPA Step 3
Identify Unsafe Control Actions
List of UCAs
STPA Step 4
Identify Loss Scenarios
List of Loss Sceanios
List of Safety Requirements

STPA Guide

Authors:	Josepha Berger
Confidentiality:	VTT Public
Version:	2.1.2024

Report's title STPA Guide	
Customer, contact person, address	**Order reference**
Project name Future Electrified Mobile Machines	**Project number/Short name** 130331 / FEMMa
Author(s) Josepha Berger	**Pages** 43
Keywords **STPA, Safety analysis method, Systems theory, Risk Priority Number**	**Report identification code** VTT-R-00848-23

This guide leads through the System-Theoretic Process Analysis (STPA) step by step. While it especially addresses safety engineers with their efforts to design and assess complex, socio-technical systems everyone interested in learning how to conduct STPA will benefit. Throughout, a continuous example from the autonomous work machine domain illustrates the STPA process. The guide introduces best practices from literature and personal experiences, highlighting deviations to complement the official STPA Handbook effectively. Further, it introduces optional extensions to STPA, that allow a focused analysis of safety and security, and human machine interactions. The Risk Priority Number approach is introduced as a method to identify the most critical results of STPA.

Confidentiality	VTT Public

Tampere 2.1.2024

Written by	**Reviewed by**
Josepha Berger, Research Scientist	Risto Tiusanen, Senior Scientist

VTT's contact address
Visiokatu 4, 33101 TAMPERE

Distribution (customer and VTT)

{Customer, VTT and other distribution. In confidential reports the company, person and amount of copies must be named. Continue to next page when necessary.}

Approval

VTT TECHNICAL RESEARCH CENTRE OF FINLAND LTD

Date:	03 tammikuuta 2024
Signature:	DocuSigned by: 1F65932542874A1...
Name:	Johannes Hyrynen
Title:	Lead, Low carbon and smart machines

Preface

Josepha Berger, Risto Tiusanen, Timo Malm

This report is related to the work done in VTT in the Business Finland funded “FEMMa” research project in its task 4.2 “System-theoretic approach for the identification of autonomy and electrification related risks”.

Tampere, 2.1.2024

Josepha Berger

Contents

1. Acronyms and Abbreviations

Table 1. STPA relevant terminology. [1], [2]

Term	Definition
Loss	– describes the result of an accident, mishap, or adverse event – describes any unwanted effect of hazard – the goal of STPA is to prevent losses
Hazard	– a system state that can be managed if eliminated or controlled in the system design – a precursor state to a loss that system designers do not want to happen
Safety Constraints, SC	– must be enforced to prevent hazards and losses – another suitable term is safety goals
Control Structure	– hierarchical representation of commands and feedback that system elements transmit to each other – system elements with higher power of command are positioned closer to the top of the control structure – consists of multiple interconnected control loops
Control Loop	– interconnected system elements within the control structure, where control actions (represented by downward arrows) and feedback (represented by upward arrows) flow between them
Controller	– a system element that transmits control actions and executes them (through actuators) to a controller with less authority or to a controlled process – a system element that receives feedback (transmitted by sensors) – utilizes the control algorithm and process model to decide which control action is given when
Control Action, CA	– a command that a controller gives to another controller or to a controlled process – depicted by a downwards arrow
Control Algorithm	– maintains continuous information about controlled processes, external system components and the environment and based on that generates control actions – for humans, control algorithms can be called operating procedures or decision-making rules
Process Model	– represents the controller's beliefs about a state and therefor is used to make decisions – may be updated by feedback – for humans, process model can be called a mental model
Actuator	– Receives and executes control actions from controller and consequently changes the process state
Sensor	– Detects information and provides feedback about the current state of the system
System element	– parts of a system, that interact with each other

Term	Definition
	– typical system elements are controllers, controlled processes, control actions and feedback, but also actuators and sensors
Feedback	– within the control structure feedback streams are visualized by upwards arrows, typically from system elements with lower hierarchical power – by labelling the arrow the information which is transmitted is made visible

2. Introduction

This guide leads through the System-Theoretic Process Analysis (STPA) step by step. While it especially addresses safety engineers with their efforts to design and assess complex, socio-technical systems everyone interested in learning how to conduct STPA will benefit. Throughout, we use a continuous example from the autonomous work machine domain to illustrate the STPA process. Our goal is to offer an alternative to the almost 200 pages detailed official STPA Handbook [1].

We believe that both current and future STPA practitioners will appreciate clear, concise, and straightforward instructions. This guide addresses some of the shortcomings [3] in the STPA Handbook, drawing from our experiences. For instance, we refined the process for identifying Loss scenarios, a step that the STPA Handbook describes rather vaguely. Our guide introduces best practices from literature and personal experiences, highlighting deviations to complement the Handbook effectively. In essence, it serves as a valuable companion to enhance your understanding and application of STPA.

Literature [4] reveals a growing interest in applying STPA across various sectors. Notably, STPA practitioners in automotive and aviation industry find significant support in the instructive nature of standards [5], which are specifically tailored to their needs. Standards, equip safety experts in these industries with effective tools, facilitating a robust application of STPA. However, a gap exists for safety experts in other sectors as they currently lack tailored guidance for implementing STPA within their specific domains [6]. While not catering to any specific industry, we emphasize the value of applying STPA in the context of autonomous operation and aim to facilitate its application in the domain.

Traditionally, ensuring the safety of automated processes involved isolating machinery from human contact. However, the drive for enhanced productivity is pushing automated operations to become more adaptable. This results in scenarios where autonomous machines coexist with non-movable machinery, manually operated machines, and human workers in the same workspace [7]. STPA can grasp the non-linear and complex interconnections of system components, making it a well-suited tool for defining both system and safety requirements [1].

3. What is STPA

In the early 2010s, Leveson at MIT (Massachusetts Institute of Technology) published a new, qualitative safety analysis method, the STPA, which addresses system-based hazards [8]. The Systems-Theoretic Accident Model and Processes (STAMP) forms the underlaying model for STPA and includes non-linear relationships of technical and organizational structures, design and requirements flaws, as well as dysfunctional interactions between components that themselves act as intended [9], [10]. The systems theory, the base of STAMP and STPA, goes beyond the assumption that unwanted scenarios are a result of a simple chain of directly related failure events or component failures. Instead, it treats a system with its complex processes and interactions among other system components as a whole, and not as sum of its parts. A system has dynamic properties in which system elements engage and impact each other. When these components interact new, non-obvious ways of how accidents can emerge arise.

Consider the example of an autonomous vehicle operating in an isolated area at 30 km/h. Although the system designers perceive this speed as safe due to human segregation from the operational environment, inclement weather conditions can transform the roads into slippery surfaces, lengthening the braking path and potentially causing accidents if the speed is not adjusted. These accidents occur without any actual machinery components failing, highlighting the inadequacy of examining system components separately and in isolation, a common practice in many traditional hazard analysis methods like fault tree analysis (FTA), failure modes and effects criticality analysis (FMECA), event tree analysis (ETA), and hazard and operability analysis (HAZOP) [1].

STPA, treating safety as a dynamic control problem is tailored for complex systems and exposes unsafe control commands that one system element could have upon another. It recognizes that losses in such systems may not necessarily stem from component failures but rather result from unpredictable and undesirable interactions among system elements. It is worthwhile mentioning that STPA uses a specific set of terminology, which might require some adaptation to those familiar to traditional tools [1].

Integrating STPA early into the development phases of autonomous machinery systems allows to identify safety critical requirements for the system proactively. It allows to address potential safety concerns in advance, instead of reacting on them by conducting system changes afterwards. Therefore, we recommend utilizing STPA to supplement traditional safety engineering activities like the Preliminary Hazard Analysis (PHA), Operating Hazard Analysis (OHA) and HAZOP during conceptual design, system design and detailed design (Figure 1) [7].

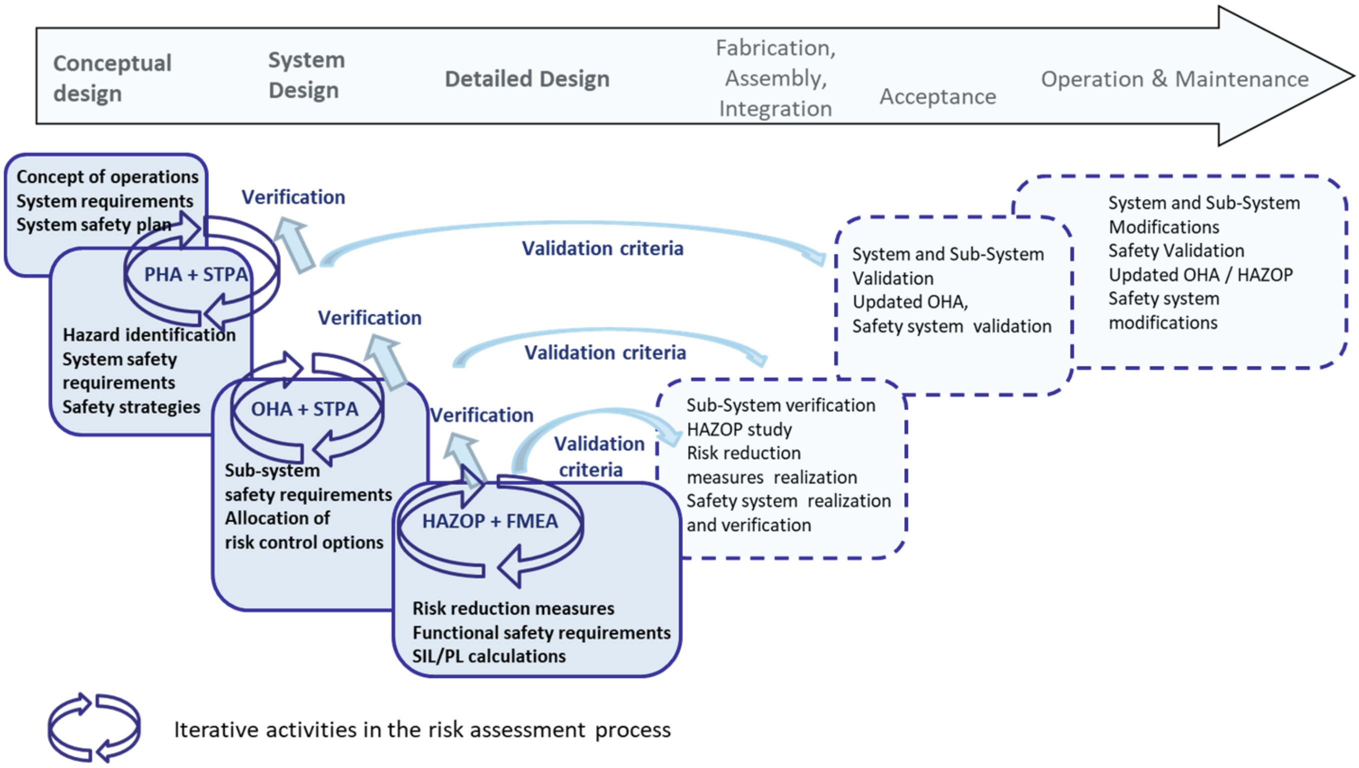


Figure 1. Safety engineering activities linked to different development phases of autonomous machinery systems. [7]

When allocating resources for the application of STPA to a specific use case, we recommend, that the team is led by a STPA practitioner, and further composes of at least one system expert strongly familiar with the system under analysation. Be prepared to involve further system component experts if the use case requires to be narrowed down in more detail. The extent of required additional resources and skills will depend on whether you are utilizing STPA for either conceptual or detailed system design and planning, or for the safety assessment and enhancement of already existing systems. While [1] recommends that STPA is most effectively conducted as a small group exercise, our experience indicates that the predominantly text-based format of the documentation can, at times, hinder the efficiency of collaborative efforts [3]. Given this, we propose that the STPA practitioner takes on the writing tasks, while other group members actively contribute through brainstorming and cross-checking the STPA reporting afterward. To enhance effectiveness, we advocate for face-to-face sessions, particularly during steps 1 and 2.

Figure 2. gives an overview of the four fundamental steps of STPA. It includes sub steps and the output they produce. The first step requires to define the purpose of the analyses as well as the system under investigation. In step 2 the hierarchical control structure is modelled to represent control actions (CA) and feedbacks that system elements transmit to each other. Step 3 identifies ways in which the CAs could cause harm and turn into Unsafe Control Actions (UCA). The successful complementation of STPA will result in a list of Loss Scenarios, revealing worst-case combinations of flawed processes and mental models, UCAs, and operational contexts that may lead to undesired situations. These Loss Scenarios are valuable if turned into system specifications and safety requirements. Therefore, they should be presented to and discussed with stakeholders involved in the system's design, operation, and safety analysis. This typically includes management, safety engineers, system designers, project managers, and other decision-makers who are responsible for ensuring the system's safety and functionality [1], [11].

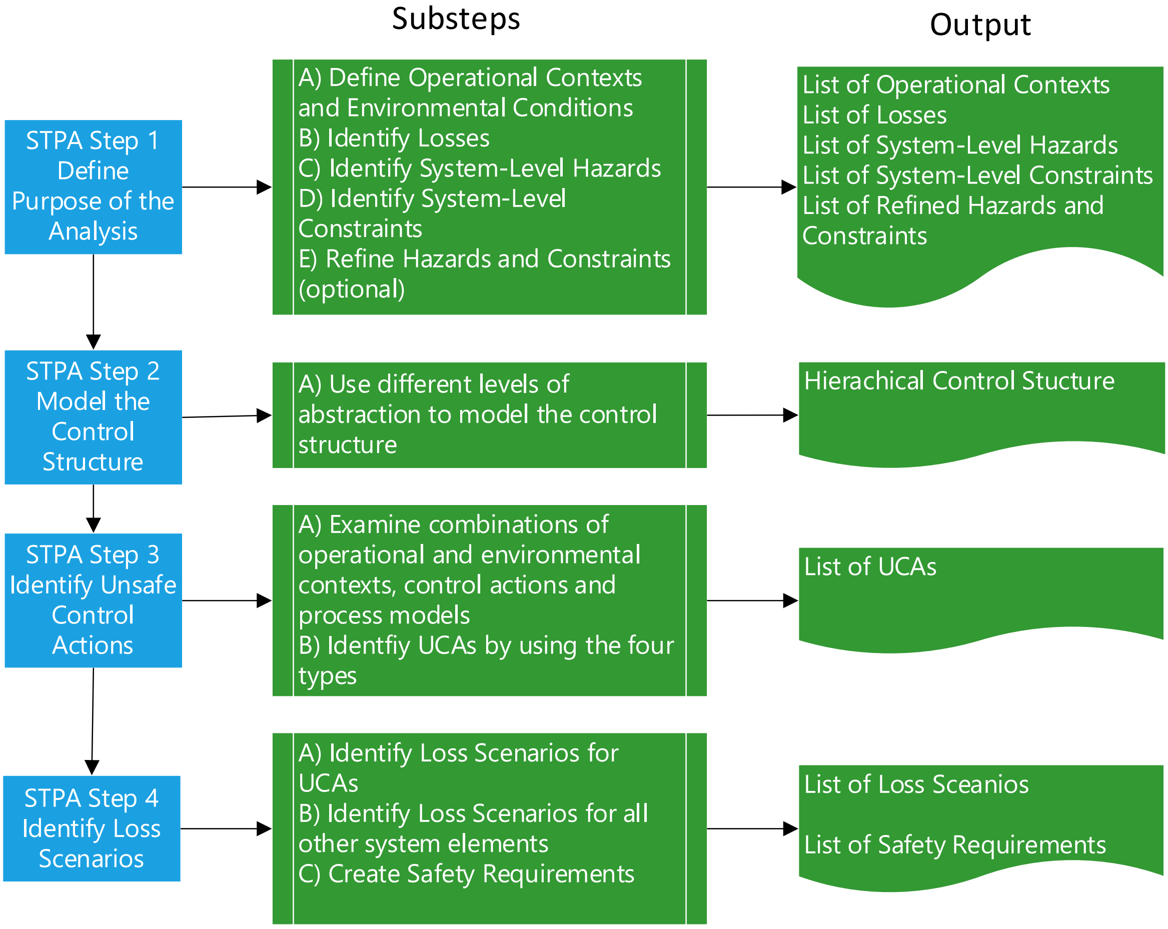

Substeps
Output
STPA Step 1 Define Purpose of the Analysis
A) Define Operational Contexts and Environmental Conditions
B) Identify Losses
C) Identify System-Level Hazards
D) Identify System-Level Constraints
E) Refine Hazards and Constraints (optional)
List of Operational Contexts
List of Losses
List of System-Level Hazards
List of System-Level Constraints
List of Refined Hazards and Constraints
STPA Step 2 Model the Control Structure
A) Use different levels of abstraction to model the control structure
Hierachical Control Stucture
STPA Step 3 Identify Unsafe Control Actions
A) Examine combinations of operational and environmental contexts, control actions and process models
B) Identfiy UCAs by using the four types
List of UCAs
STPA Step 4 Identify Loss Scenarios
A) Identify Loss Scenarios for UCAs
B) Identify Loss Scenarios for all other system elements
C) Create Safety Requirements
List of Loss Sceanios
List of Safety Requirements

Figure 2. STPA procedure and output. Adopted from [12]

4. Applying STPA

4.1 Step 1: Define Purpose of the Analysis

Step 1 of the STPA analysis deals with describing the system or use case that shall be analysed. During this phase you will set the boundary of the analysis and gather and create relevant documentation needed later in the process. The following sub steps will take you through all that is required to defining the purpose of the analysis and result in a conceptual use case as depicted in Figure 3.

Environment
System boundary
Use Case

Figure 3. Example of a conceptual use case. Modified from [1]

4.1.1 System Scope and Boundary

At first you are required to define which use case you want to analyse with STPA. If this is your first time practicing STPA it is advisable to start with a simple case, because even a narrow system can lead to an extensive analysation process. Therefore, if necessary, you can isolate your case from the overall system.

In reality it might not make sense to fully extract your use case system from its overall operations, because it would rib the use case out of its context and make the analysis invalid. You can overcome this challenge by flexibly defining the system boundary. For example, if reasonable you may neglect certain inputs or effects from the overall system to the subsystem / your use case system and vice versa. Another option to avoid escalation of the analysis is to agree on a specific type of impact to the case system or to a particular case system element. Most important factor is that decisions on the analysis boundaries and possible external interactions have been made consciously and with system experts' advise. If you realise throughout the analysation process, that your use case is lacking relevant system elements you can expand your system by taking advantage of STPA's iterative nature, that allows adjustment of the original analysis.

The following list presents a few examples on how to narrow down your use case:

- exclude special operations like maintenance, restart, or shut-down scenarios
- separately perform an STPA on precisely such procedures
- reduce the number of dynamic elements like vehicles, pedestrians, personnel, or amount of machines
- focus only on certain operational conditions (e.g. dry, daylight, heavy rain, high level of operational activity)

In general, STPA is considered a worst-case analysis method, which means it doesn't analyse existing safety measures or safeguards when applied on existing systems. These are neglected because in a worst-case scenario these safety features might not work as expected, or they might not be enough to prevent hazards. [1] However, STPA is suitable to analyse the effects of safety measures themselves.

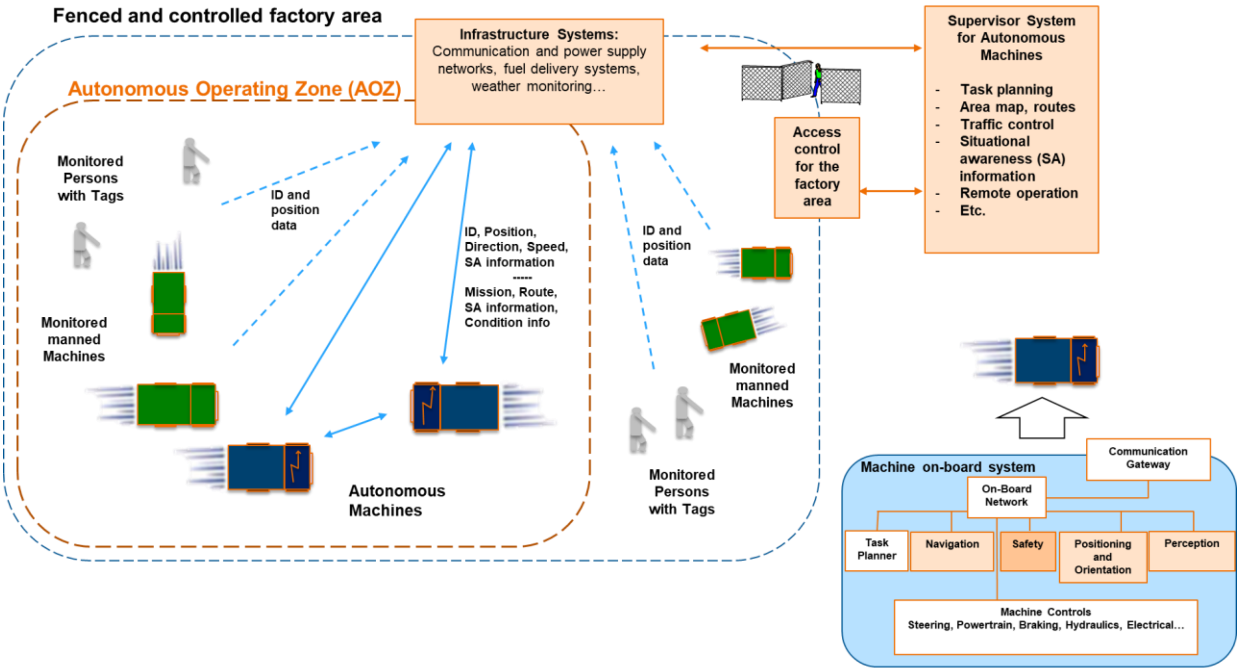
Especially, if the safety measures consist of CAs, that are transmitted in hazardous situations to prevent losses and therefor serve as safeguards.

It is beneficial for the subsequent analysis to document the key points of your use case and the conclusions of discussions about how to handle interactions that are outside the use case. Typical outputs of this sub step can be conceptual images of the use case situation, textual use case description, lists or memo's of your workgroup's discussion. As already mentioned, these documents can be edited and merely serve your team as reminder of what you agreed on.

Example: System Scope and Boundary

The overall system of which the use case example is part of is described below[1]. It facilitates the understanding of the use case that serves as a continuous example in this guide. Once the overall system is described, the system scope and boundary for the use case are defined.

This example's overall system consists of an autonomous machine (AM) including an automated mobile machine system (onboard sensor system and some situation awareness information) and its operating environment. An outline of the overall case system with its operating environment is shown in Figure 4. Different traffic situations on the route of the AM are illustrated Figure 5. The characteristics of the overall system can be taken from Table 2.


Fenced and controlled factory area
Infrastructure Systems:
Communication and power supply networks, fuel delivery systems, weather monitoring…
Autonomous Operating Zone (AOZ)
Supervisor System for Autonomous Machines
- Task planning
- Area map, routes
- Traffic control
- Situational awareness (SA) information
- Remote operation
- Etc.
Access control for the factory area
Monitored Persons with Tags
ID and position data
ID, Position, Direction, Speed, SA information

Mission, Route, SA information, Condition info
ID and position data
Monitored manned Machines
Monitored manned Machines
Autonomous Machines
Monitored Persons with Tags
Machine on-board system
Communication Gateway
On-Board Network
Task Planner
Navigation
Safety
Positioning and Orientation
Perception
Machine Controls
Steering, Powertrain, Braking, Hydraulics, Electrical…

Figure 4. An outline of the overall case system and its operating environment.

[1] This use case is adopted from VTT's 2023 Customer Report "New EU requirements and safety engineering methods for autonomous machinery systems".

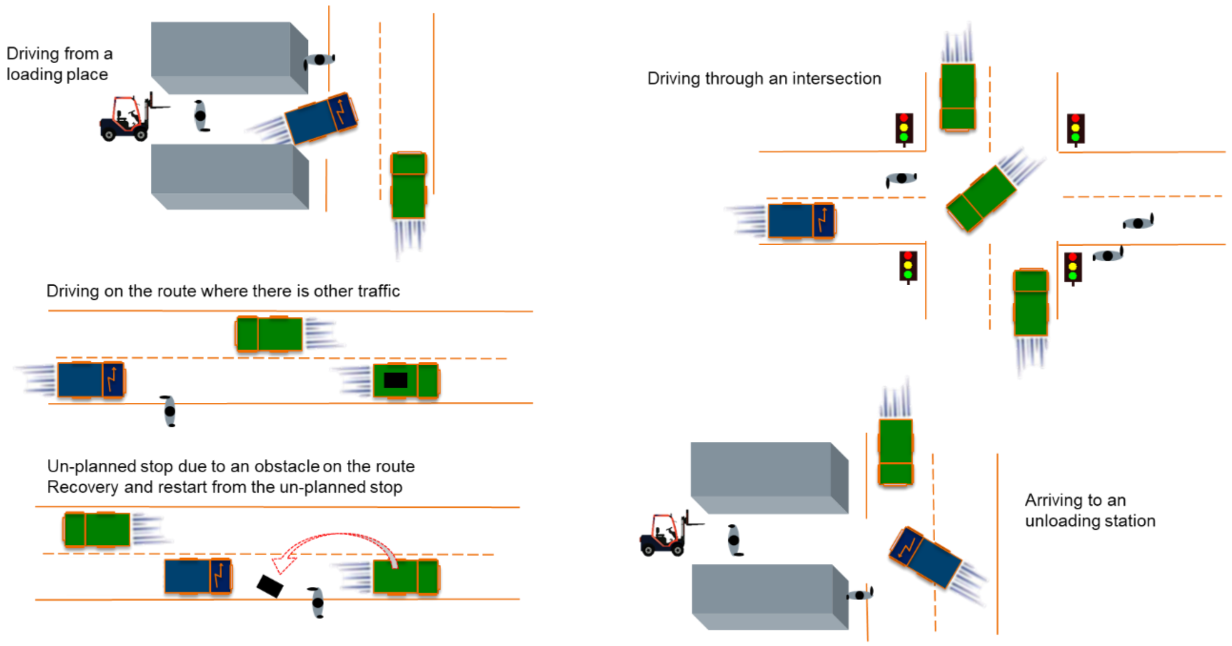

Driving from a loading place
Driving through an intersection
Driving on the route where there is other traffic
Un-planned stop due to an obstacle on the route
Recovery and restart from the un-planned stop
Arriving to an unloading station

Figure 5. Different traffic situations on the route of an AM.

Table 2. The overall system characteristics and the main functionalities.

System characteristics	**Overall system descriptions**
Overall purpose of the system	AM transports goods within an AOZ of an industrial area.
Basic functions of an autonomous machine	The basic functionalities of the autonomous machine are not the subject of interest. It is assumed that they (e.g. sensors related to automatic driving, motion control, stop functions, etc.) work correctly and without interference and meet safety requirements.
Driving speed on the route	The driving speed of an AM is 20 km/h – 30 km/h NOTE! – ISO 22737:2021 Intelligent transport systems — Low-speed automated driving (LSAD) systems for predefined routes — Performance requirements, system requirements and performance test procedures, maximum speed is 32 km/h, – ISO 5010:2019 Earth-moving machinery — Wheeled machines — Steering requirements, 20 km/h
Environmental conditions	The AM operates outdoors, in Nordic conditions
Machine dimensions	The starting point for autonomous driving is that the driving route and the dimensions of the machine are known – the AM can drive on routes designed for truck traffic. – the dimensions of the AM may change, e.g. when lifting or moving a load. E.g., due to the shape and size of the load, there may be blind spots for observing the AM’s surroundings The combined maximum mass of the AM and the load is known to be approx. 30 t, which affects e.g., stopping distance
Operational area and infrastructure	The case system works in an industrial area with restricted access. Within this, the operating range of the AM is not physically limited. Crossing areas may have limited visibility to crossing traffic.

	There may be sensors related to safety functions along the driving route, e.g. cameras in the crossing area and ID Tag reader devices at certain points along the driving route of the AM. Information about the conditions in the area can be obtained from e.g. weather condition or route condition sensors: – slippery road surface, visibility, wind speed
Perceptiveness	Safety sensors onboard the AM detect an obstacle or a person at a maximum distance of 4 m LIDAR and RADAR sensor systems onboard related to machine automation and environmental observation –
Decision-making ability, foresight	The AM can first slow down when it detects an obstacle on the route. If the obstacle is not removed, the machine stops The AM can go around the obstacle and return to the original route, if it can be done safely. However, the machine does not plan a completely new route. The need for the ability to create a map of the operating environment or define new routes is assessed in connection with the analysis e.g. Simultaneous localization and mapping (SLAM)
The responsiveness of the machine	It is assumed that the AM is able to brake and stop within the observation distance, taking into account the factors affecting the stopping distance. It is assumed that the minimum stopping distance at maximum speed in good conditions (dry asphalt) is approx. 15 m NOTE! – ISO 3450:2011 Earth-moving machinery — Wheeled or high-speed rubber-tracked machines — Performance requirements and test procedures for brake systems. In good conditions (dry asphalt), when mass < 32 tons, the minimum requirement is 18.8 m
Localisation	The AM has: – location information, route information, digital map, information about the area – Absolute knowledge and relative knowledge – Assumed positioning accuracy e.g. 10 cm – GNSS, IMU (accelerometers, gyroscopes), odometer
Other traffic in the area	People and other traffic in the industrial area, where access is limited, have tags. However, one cannot be sure of tag's presence. The range of the tag is short (passive tag – the range of the reader device is a few meters; active tag – the range is 10–20 meters, even in good conditions less than a hundred meters). Vehicles could have more effective active tags than persons. The AM does not know in advance the characteristics of other vehicles moving in the area (size, shape, cargo).
Traffic rules	Traffic at intersections is controlled by traffic lights that prioritize automatic traffic. Pedestrians may only cross the road at a crosswalk equipped with traffic lights. Pedestrians have the opportunity to control traffic lights to safely cross the road. It is assumed that manual drivers obey traffic rules and comply with the provisions of the Road Traffic Act. Pedestrians and cyclists can act in traffic unpredictably and unexpectedly.
Mission planning and ensuring correctness	The AM receives the mission through the process control and the traffic control system. The AM has the ability to ensure that it is on the planned route.
Communication network connections	When the network connection is interrupted, the AM goes into safe mode and stops if necessary. The AM is allowed to continue the mission if the network connection is interrupted, if according to the risk assessment it can be done safely depending on the traffic situation and the point of the route. NOTE! – According to current requirements [30], the continuation of automatic operation can be allowed based on a risk assessment.

Communication with other traffic	The AM clearly indicates its operating status to other traffic (signal sounds and warning lights, etc. information means
Traffic and Mission control system	The traffic and mission control system reserves areas / parts of the route for one AM at a time. The AM has the right of way on the route.
Supervisor / operator	An AM system has a supervisor / operator who supervises the operation of the system and assigns missions to the AMs. – The supervisor does not remotely control individual machines. – According to the new machine regulation being prepared, the supervisor is a 'driver'
Remote controller and remote control system	The system has a separate remote controller. The AM can be controlled remotely using a camera image. The remote operator can steer the autonomous machine, e.g., to the side of the carriageway, away from the driving lane. In remote control, the machine's safety functions are turned on in the same way as in autonomous driving. The remote operator has access to video from surveillance cameras in the area.

It is possible that either the overall system of which your use case is part, is already given to you by your commissioner or you need to work it out together. If the latter is applicable, then Table 2 above gives an idea to which extend you might need to clarify details before officially starting with the STPA.

Now, after the overall system has been clarified, we can move to the actual STPA analysis and define the use case system and its boundaries. Therefore, the following points describe which situations and elements are considered and which neglected:

- The effects of a human element, functioning as system supervisor of the Traffic and Mission control system are of interest and shall be analysed.
- Interactions between the Traffic and Mission control system, one exemplary AM, traffic lights and other traffic and items are of interest and shall be analysed.
- The safety functionalities of the AM system are not considered.

This use case is purposefully set to be very narrow to keep the example comprehensible.

4.1.2 Operational Context and Environmental Conditions Identification

A system or elements of a system might be exposed to varying environmental conditions or applied in different operational contexts. Below you find a few examples on how operational context and environmental conditions could impact the functionality, and consequently the safety of your system [1], [2], [13]:

Examples of how Environmental Conditions can impact safety:

- Braking distance of a vehicle might be different on a dry road compared to rainy road conditions
- Sunlight can affect the performance of a video camera
- Sensors can be impaired during night and snowfall

Examples of how Operational Context can impact safety:

- A sharp break command might be unsafe on a high-speed motorway but save lives on a productions site road
- Sensors of an AM might have difficulties transmitting data while operating underground
- Physical separation of human from a machinery might allow the machinery to have more functionalities without endangering human safety

Safe operation must be granted in all different foreseeable conditions and therefore their impacts on the system must be understood. STPA has the feature to consider the effects of varying conditions on a system, its elements, and their interaction. In order for the STPA analysis to do so it is important that you are aware of and list the operational contexts and conditions your use case is exposed to or agree under which conditions operation must be prohibited.

The list below provides an expandable collection of operational contexts and environmental conditions which might be relevant.

Table 3. Selection of Operational Contexts and Environmental Conditions.

Operational Contexts	Environmental Conditions
site infrastructure	weather conditions
gravel road	water alongside road
road signs / traffic lights	shadows
sidewalk	sun
cross section	temperature
road merging	time of day
road branching	air humidity
uphill	ice
downhill	fog
mixed fleet	
underground	
speed	

This step is an addition to the original step 1 described in the STPA Handbook. However, also the STPA Handbook stresses the importance of contexts a system can be in, but rather includes them in the identification of UCA, step 3 of the analysis. The suggestion to identify operational context and environmental conditions already in this phase is based on personal experience and recommended by [2]. Finding relevant and realistic conditions of your own use case supports in setting the boundaries of the analysis (step 1), modelling the control structure (step 2) and in avoiding missing UCAs and Loss Scenarios that might only occur in certain combinations of conditions.

You should involve system experts to help list relevant aspects to your use case. Depending on how isolated your use case is from the outside world and the complexity of your operational context this list might even be very short.

Example: Operational Context and Environmental Conditions

For our use case we assume the following operational context and environmental conditions:

- The AM operates outdoors in an isolated industrial area.
- Pedestrians within the same operating area are aware of the autonomous function of the AM.
- Weather conditions are considered very generally. Analysing the impact of weather is not main part of this analysis and therefore the impact of Nordic conditions is neglected.
- It is assumed that AM operates during daytime (no night or dusk/dawn conditions).
- There is only one AM moving within the same area.
- The operational area contains manually controlled machines and workers with identification tag.
- It is possible, that some obstacles or persons are not carrying a tag.

4.1.3 System-Level Loss Identification

According to the Handbook the identification of losses is the first official substep of STPA. Losses do not need to be solely associated with safety [2]. Basically, any value or goal can be translated into a loss. The loss of such would be unacceptable to the stakeholders and must be prevented. It depends on the level of depth of your use case, but typically when identifying losses, it is not necessary to create an overly detailed list. A more general formulation can suffice to summarize incidents of a particular type. For instance, the formulation "Loss of life or injury" can effectively encompass all types of injuries, that are severe enough to impact the system goal. Typical losses are:

- L-#: Loss of life or injury to people
- L-#: Loss or damage to system objects (vehicles, tools, machinery, etc.)
- L-#: Loss of mission (the goal of the use case or overall system)
- L-#: Loss of customer satisfaction
- L-#: Environmental loss
- L-#: Loss of sensitive information (especially relevant if you consider cyber security)
- L-#: Loss of reputation

One of STPA's strengths is the traceability between its various outputs. STPA's traceability allows, to draw the connection from a hazard to a loss or from a loss scenario to an UCA and vice versa. You can achieve this traceability by labelling your losses (L-1, L-2, …L-#) and refer to them by putting them in brackets at the end of a description ([L-1], [L-2], […]). For examples see 4.1.4.

Example: System-Level Loss Identification

For our use case we identified the following losses:

- L-1: Loss of life or injury to people.
- L-2: Loss of, or damage to vehicle (AM or others).
- L-3: Loss of, or damage to objects outside the AM.
- L-4: Loss of mission (production).

4.1.4 System-Level Hazard Identification

In STPA, hazards are defined as system states or conditions prior to an accident, or more precisely, in STPA terminology, before a loss occurs. Note, hazards in the context of STPA rather refer to an unsafe action, condition or command that could result in a loss and shall not be confused with the long-established understanding of hazards induced by physical entities, like crushing hazard, cutting hazard [30].

As STPA is about controlling the system, its elements and how they affect one another, hazards due to component failures are not considered. Further, hazards caused by environmental impacts are only considered for system states that are within the control of the system designer. Examples of factors that are within the control of the designer are to adjust speed limits according to road conditions, to establish a strong Wi-Fi connection that ensures a stable exchange of commands also underground, or to enhance system elements to withstand inclement weather. Hazards, that are typically beyond the control of system designers are for example intentional misuse of the machine / system or natural disasters.

The handbook [1] suggests the following syntax to describe hazards:

<Hazard specification> = <System> & <Unsafe Condition> & <Link to Losses>

where, <System> refers to any element of the system, and <Unsafe Condition> describes the state or condition that must be prevented for the system element. By putting the respective loss(es) in brackets behind the hazard the <Link to Loss> is established. It is important, that the hazard description contains these elements, but the order does not matter. Phrase the descriptions to form sentences that make sense. The output of this step is a list of hazards, that describe a state or condition that, if not prevented, leads to a loss in your use case. Some examples of hazard descriptions gathered from different STPA analysis are [1], [2], [12], [14]–[16]:

- H-#: Object violates minimum distance to another object
- H-#: Machine moves when people are nearby
- H-#: Equipment stops unintendedly during operation
- H-#: Actuators activate when there are people nearby
- H-#: Exposure to toxic materials above safe level
- H-#: Inability to remote control
- H-#: Unmanned aerial vehicle does not complete surveillance mission
- H-#: Ship sails in too shallow water
- H-#: Gas compression system continues to supply gas when gas leaks to the environment

Example: System-Level Hazard Identification

For our use case we identified the following hazards, and linked them to the previously identified losses accordingly:

- H-1: The AM moves when there are people nearby. [L-1][L-4]
- H-2: The AM moves when there are vehicles or other objects in the way. [L-2][L-3][L-4]
- H-3: The AM leaves the area where it is intended to operate. [L-1][L-2][L-3][L-4]

4.1.5 Define System-Level Safety Constraints

For simplicity, this guide refers to System-level safety constraints as safety constraints (SC). Another suitable and rather self-explanatory term for SC is the term “safety goal” [2]. This term emphasizes the fact, that SC shall be seen as safety objectives, that declare which situations must be prevented and shall not happen. Seen from a high-level, SCs appear like trivial restatements of hazards as constraints on the design to ensure that hazards do not happen. This simplification might be true for conceptual, high-level use cases but does not justify the actual importance of this step: In step 3, when the actual analysis starts, we compare CAs against these safety objectives and identify ways when these safety objectives are violated.

To formulate a high-level SC, the hazard description is inverted into a hazard-free condition which describes the necessary system conditions or behaviour to avoid the hazard and consequently the loss. By providing a link to the hazard it is easy to follow due to which hazard a certain SC must be provided. The following list gives some examples of this practice [11], [15], [16]:

- SC-#: Machine must not move when people are nearby [H-#]
- SC-#: Equipment must be available to work as intended [H-#]
- SC-#: The ship must not sail in too shallow water [H-#]
- SC-#: Object must maintain minimum separation distance [H-#]
- SC-#: Gas compression system must stop compressing gas when gas leaks to the environment [H-#]
- SC-#: People must not work in conditions where they can be exposed to toxic materials above safe level [H-#]

Note, to meet the demands of advanced use cases, multiple SCs might be required to prevent one hazard.

Example: Define System-Level Safety Constraints

For our use case we identified the following SC, and linked them to the previously identified hazards accordingly:

- SC-1: The movement of the AM shall not cause safety risk to people nearby. [H-1]
- SC-2: The movement of the AM shall not cause risk of collision when there are vehicles or other objects in the way. [H-2]
- SC-3: The AM shall not leave the area where it is intended to operate. [H-3]

4.1.6 Refine Hazards and Constraints (optional)

Refining hazards means to break down the previously defined hazards into more detailed descriptions by following the same syntax. The output will be a list of sub-hazards that can be used to create more specific safety constraints. This step is optional but has the potential to add a more detailed perspective to designing your system. Especially if you analyse large and complex systems it is beneficial to break-down hazards and approach them more directly. [1], [17]

Example: Refine Hazards and Constraints

The example below (Table 4) shows a set of sub-hazards which have been refined based on hazard H-1. It is possible to refine these constraints to even more detail and for example define the radius in which the AM must adjust its driving to human presence.

Table 4. Sub-hazards derived from System-level hazard identification-example H-1 and their specific constraints.

Sub-hazards for H-1	Constraints related to the refined hazard
The AM does moves near people when they cross the road at cross walks.	AM must stop when people are on the cross walk.
The AM moves near people when they are on AM's path other than on the cross walk.	AM must stop when people are on its path.
Deceleration is insufficient upon spontaneous breaking.	AM must adjust its speed as soon as people are nearby, even if they are not on the road yet.

4.2 Step 2: Model the Control Structure

Step 2 of STPA encompasses the modelling of the control structure. Before carrying on reading, revise definitions important to this section (especially control loops, CAs, feedback, system element).

The STPA control structure is characterized by its hierarchical visualization of command- (CA) and feedback- flows among system elements. This means, that system elements with greater authority are drawn above those who receive and execute their commands. CAs are indicated by downwards arrows and feedbacks are indicated by upwards arrows. This interaction results in a so called control loop (Figure 6).

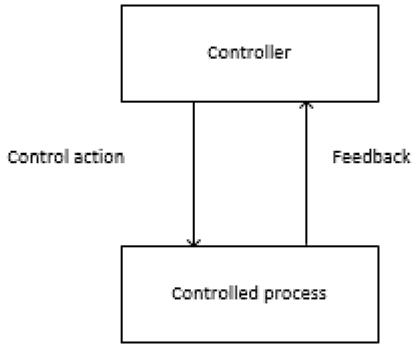


Figure 6. A generic control loop. [1]

A control structure depicts all relevant system elements and their interactions in the form of multiple such control loops. Inputs that come from outside the boundaries of the control structure but influence a system element’s CAs or feedbacks, must be added with a horizontal arrow. Label the feedback and CA arrows as accurately as possible. Ideally, the labels are a short description or keywords of the command or feedback information they are transmitting. This helps anyone working with the control structure to swiftly grasp its concept. Figure 7 provides a basic illustration of how a typical STPA control structure looks like [1] [2].

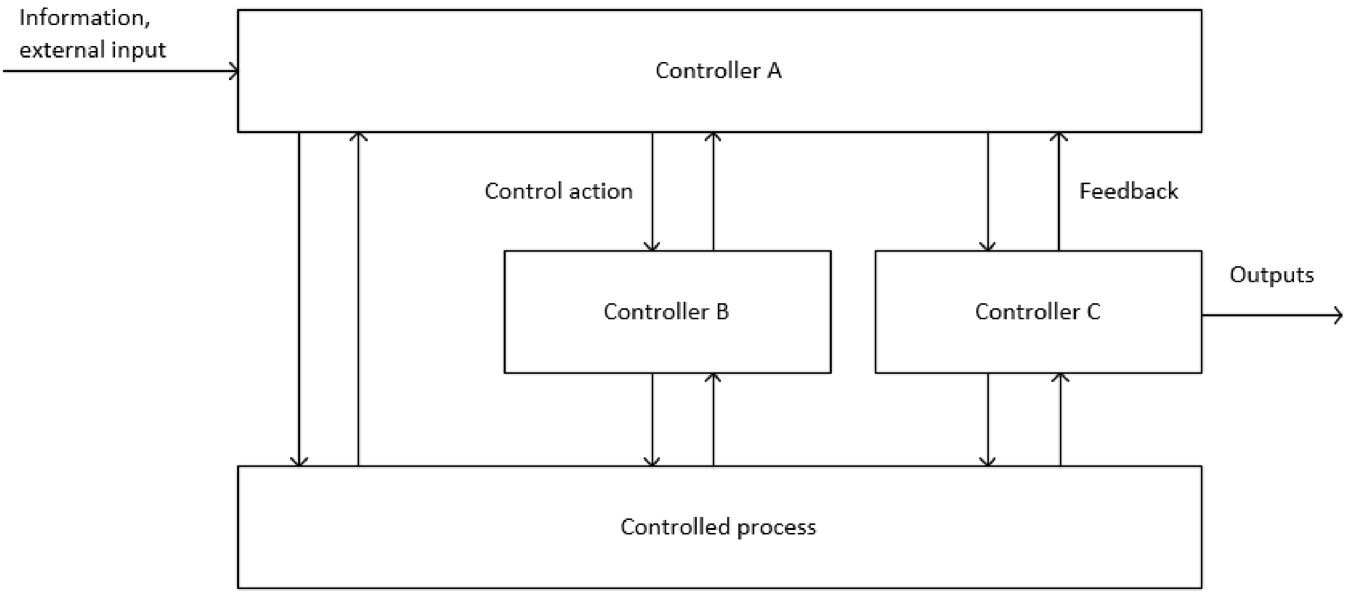


Figure 7. Generic hierarchical control structure as a set of multiple control loops. Adapted from [1]

Another approach that allows to increase readability and enhance the communication experience when discussing the control structure is to use colour coding. View Figure 8 to see how colouring CAs, feedbacks, human controllers, system controllers, controlled processes, the environment, related systems, and interactions can improve the understanding of different system elements. [2]

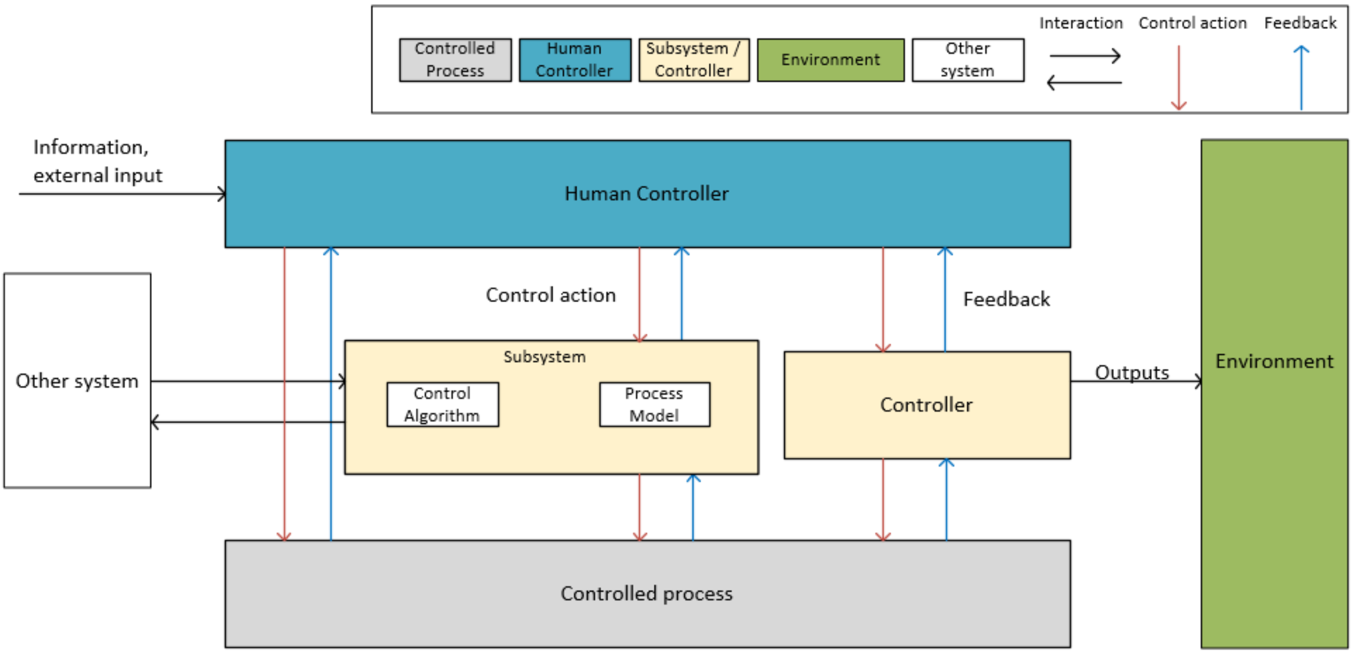


Figure 8. Colour coded control structure. Adapted from [2]

Further, to deal with complex control structures more efficiently, you could start with drawing a simplified control structure, meaning a high-level representation of the use case system. The handbook refers to this step as “abstraction”. Abstraction means to leave out certain features of your control structure, so your team would not get repelled or overwhelmed by managing this otherwise highly laborious task of modelling the whole, detailed control structure at once. Abstraction allows you to start very generally as illustrated in Figure 7 and later add details to certain elements in one or more additional rounds. Details should be added only where it suits your analysis’ scope and boundaries. You do not need to open up each system element equally. On the opposite, irrelevant system elements such as subsystems or controlled processes can be drafted on a high level, simply to show their place in the control structure. [1], [2]

By the following means detail can be added to the control structure if required:

Subsystems

One system element can encompass multiple subsystems. In a high level of abstraction, these subsystems are not shown. However, if your use case targets the analysis of certain system elements you must open them up by depicting also the underlaying subsystems and / or system elements. This results in a more detailed control structure as illustrated in Figure 9.

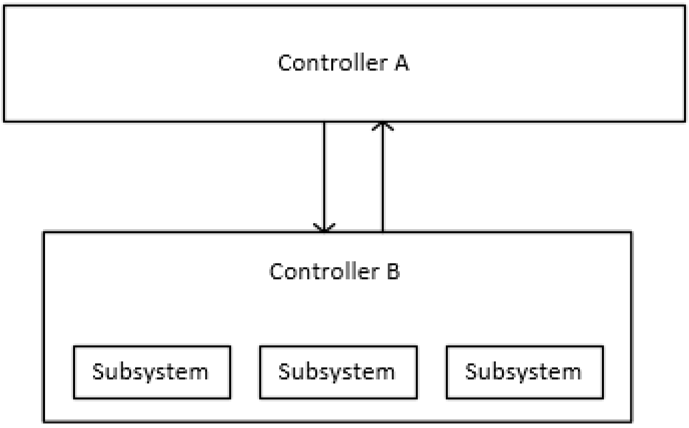


Figure 9. Refined control structure with subsystems.

Below you find literature examples, which can be revised for more details:

- A system's high-level control structure depicts an autonomous controller called "Ship". The "ship" contains controller elements like "Autonomous navigation system", "Autonomous machinery management system", "Supervisory Risk Control", "Electronic navigational charts", "Situational awareness system", and "Physical body". The STPA practitioners are interested in the Supervisory Risk Control, which is why they open up the control structure for this system element and reveal the control loop dynamics of the subsystem within this element. [18]
- A control structure of a low-speed automated driving system focuses on a controller called "Autonomous Control System". Therefore, it shows the interrelation of the sub elements (like "Global Path Planning", "Local Path Planning", "Obstacle Detection classifier", "Fleet supervisor" and "Localisation") of the "Autonomous Control System" controller. Revise [2] if interested in seeing how selected parts of the control structure are emphasized.

Process model and control algorithm

The reasons why a controller sends a specific CA are defined by its control algorithm. The control algorithm is like a set of rules that govern the decision-making process for selecting appropriate CAs. To do so, the control algorithm utilizes various factors, including the controller's process model, previous control inputs and outputs, and other relevant considerations to determine which CAs are suitable for achieving the desired outcomes. The process model represents the internal understanding or belief about the process being controlled, the controller, the environment, and general assumptions about the system state. Feedback is a typical input to update the controller's process model. Wrong assumptions about a system state due to a flawed process model can lead to inappropriate CAs which themselves can cause loss. Therefore, expanding the control structure to include a controller's process model and control algorithm helps pinpoint the inputs required for flawless operation. Figure 10 serves as an example. [1]

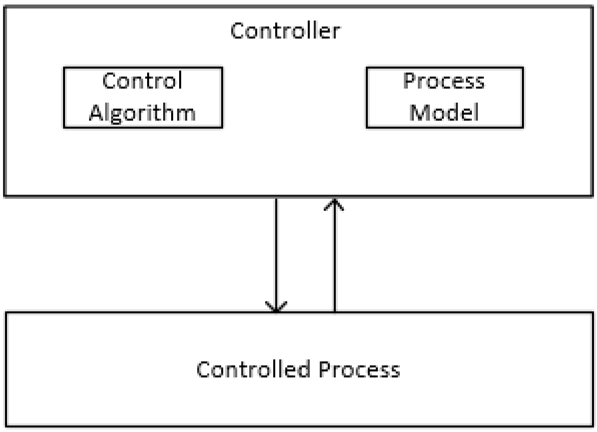


Figure 10. Refined control structure with control algorithm and process model. [1]

Such a level of refinement is not necessary unless specifically requested in the scope of your STPA analysis. However, if such an approach is desired, we recommend analysing and modelling each controller one by one. This helps in ensuring a focused and effective examination of the system's control mechanisms.

When considering the human factor, an analogy can be drawn between the controller and a human operator. In this context, the control algorithm equals decision-making, while the process model is referred to as the mental model or belief. For more details on how the human factor can be integrated into STPA, please refer to chapter 5.3.

In general, do not feel surprised when modelling the control structure feels like a learning process about your use case system. STPA enables to identify non-obvious CA paths by viewing your use case from a different angle. However, to succeed we recommend to strongly involve system experts and system designers. Also, keep in mind the iterative feature of STPA, which allows you to tweak scope and boundaries if the control structure otherwise would exhaust available resources.

Example: Model the Control Structure

Figure 11 displays the control structure of our use case, which has been modelled by using MS Visio. The AM is depicted as a set of subsystems to demonstrate interactions among various controller elements. In the context of the provided example scope, this level of detail is deemed more extensive than required. A higher level of abstraction would be sufficient. Nonetheless, this level of detail was chosen to aid the reader in generally comprehending the possibilities of step 2. and to demonstrate STPA’s flexibility and potential for exploring additional aspects of the control structure if they seem relevant.

CAs and feedback are clearly labelled, providing a concise description of the information they transmit.

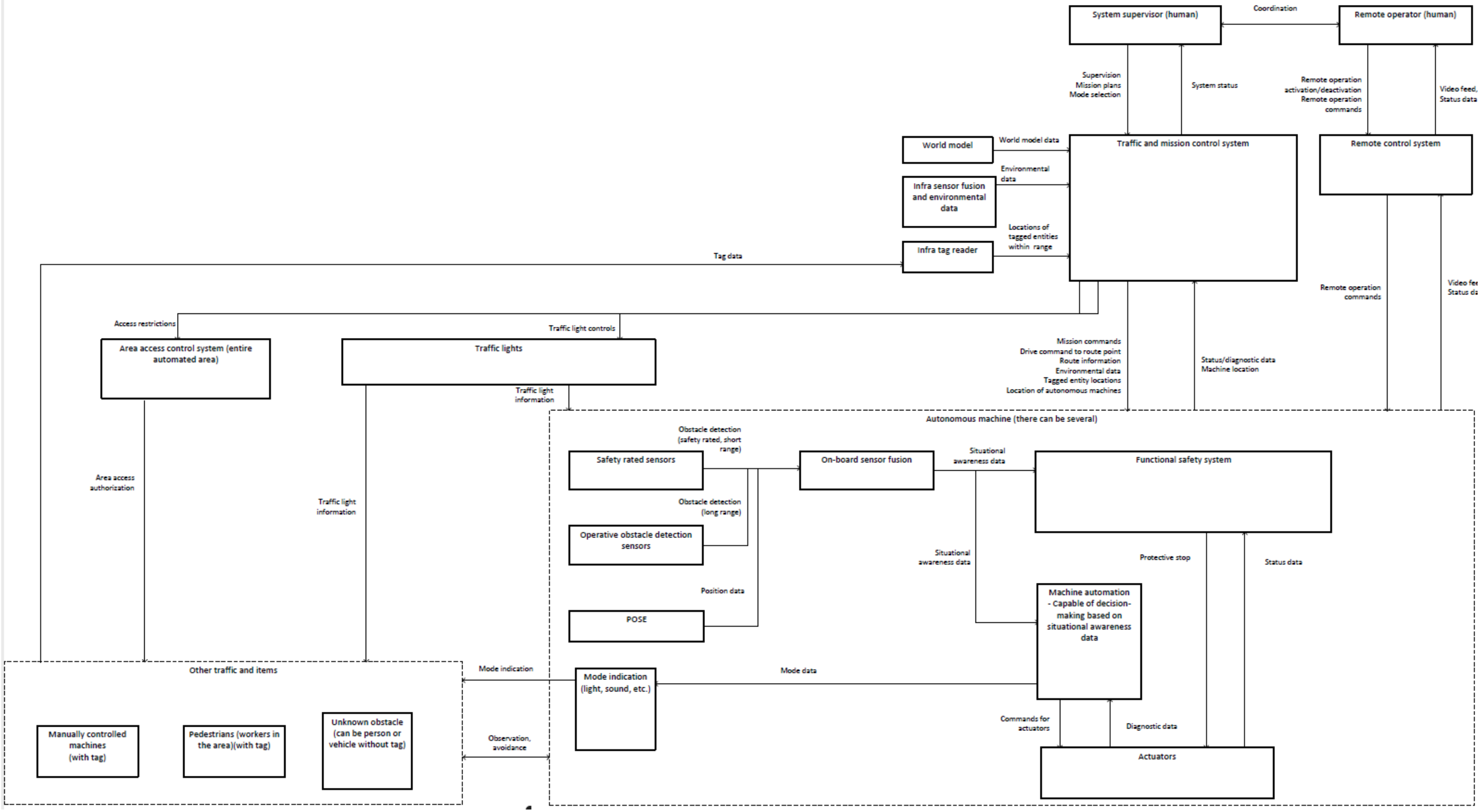


Figure 11. Control Structure of the use case.

4.3 Step 3: Identify UCAs

The STPA method assumes that losses or causes for hazards primarily origin in errors in CAs. As a result, Step 3 of STPA is dedicated to identifying Unsafe/Unsecure/Unwanted or Unexpected Control Actions (UCAs). These UCAs are derived from the CAs identified in step 2 and have the potential to violate SCs defined in step 1. In step 2, the control structure visualizes these CAs within the system (downward arrows) but does not consider the context in which they could potentially cause harm and become UCAs. According to the STPA Handbook, there are four types or possibilities by which a CA can become a UCA:

1. “providing” a CA causes a hazard,
2. “not providing” a CA causes a hazard,
3. “providing too soon, too late or in the wrong order” causes a hazard and
4. “stopping the CA too soon or applying it too long” causes hazards.

Formulating the UCAs results in written sentences that describe circumstances in which CAs become unsafe. UCAs must contain the following parts:

UCA-#: <Source><Type><Control Action><Context><Link to SC><Link to Hazard>

whereas UCA-#, labels each UCA with a number for easy reference, Source, is the system element which provides the command, Type, refers to either providing, not providing, etc., and Context, describes in which circumstance or environmental condition the CA becomes unsafe. The underlaying hazard is referred to with [H-#]. One UCA can be linked to multiple hazards. To link the SC is a deviation from the Handbook [1]. We argue that SCs are important results of STPA and practitioners benefit from tracing which SC is violated by which UCA.

To identify UCAs, proceed like this:

- Select one of the CAs which you identified when modelling the control structure.
- Contemplate about this CA. Which of the four types make this CA unsafe / violate the identified SCs? Think about all possible operational contexts and environmental conditions, that you identified in step 1. In which combination(s) have type and operational context and environmental condition the potential to turn this CA into an UCA?
- Describe the UCA by following the syntax.
- Proceed with the other CAs that are relevant for your use case. If you modelled a more extensive control structure to support your understanding of your use case you may come to the conclusion to leave out irrelevant CAs.

Example: Identify UCAs

For this example, we chose to present only two CAs that have the potential to become UCAs. However, in a complete STPA analysis, this step would involve analysing all relevant CAs. To enhance clarity and facilitate comprehension of our example, the snapshot of the
control structure (Figure 12) highlights the selected CAs in green.

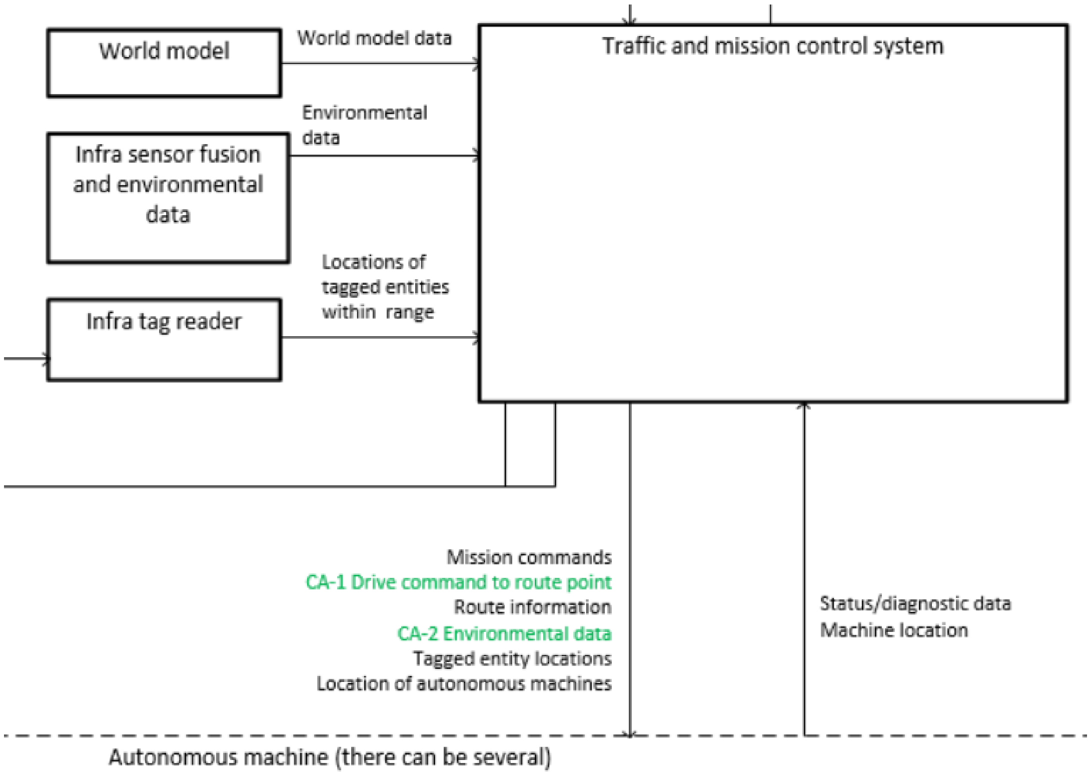

World model
World model data
Traffic and mission control system
Environmental data
Infra sensor fusion and environmental data
Locations of tagged entities within range
Infra tag reader
Mission commands
CA-1 Drive command to route point
Route information
CA-2 Environmental data
Tagged entity locations
Location of autonomous machines
Status/diagnostic data
Machine location
Autonomous machine (there can be several)

Figure 12. Exemplary CAs to be analysed in Step 3.

Figure 12 shows, that CA “Drive command to route point” is sent from the Traffic and mission control system to the Autonomous Machine (AM). The CA “Environmental Data” refers to the Traffic and mission control system’s requirement to supply relevant information regarding environmental conditions to the AM.

Table 5. Exemplary set of identified UCAs.Table 5 lists the UCAs identified for the selected CAs.

Table 5. Exemplary set of identified UCAs.

Control action (CA)	**Providing causes hazard**	**Not providing causes hazard**	**Providing too soon / too late / in wrong order causes hazard**	**Stopping too soon / applying too long causes hazard**
CA-1 Drive command to route point	UCA-1.1 Traffic and mission control system provides drive command but to an incorrect route point [H-1][H-2][H-3] [SC-3] UCA-1.2 Traffic and mission control system provides drive command to route point, but the command is not adjusted to the given operational context [H-1][H-2][H-3] [SC-1]	UCA-1.3 Traffic and mission control system does not provide drive command while AM is already on the path [H-1][H-2][H-3] [SC-3]		

	[SC-2]			
CA-2 Environmental Data	UCA-2.1 Traffic and mission control system provides drive command to route point, but the command is not adjusted to the given environmental condition while AM is driving [H-1] [H-2] [SC-1] [SC-2] (AM operates by not complying to required parameters)	UCA-2.2 Traffic and mission control system does not provide environmental data during extraordinary environmental conditions [H-1][H-2] [SC-1] [SC-2] (AM operates by not complying to required parameters)	UCA-2.3 Traffic and mission control system provides environmental data too late to adjust accordingly while AM is driving [H-1][H-2] [SC-1] [SC-2] (AM operates by not complying to required parameters)	

To enhance traceability, all UCAs originating from the same CA (e.g., CA-1) are labelled accordingly as UCA-1.1, UCA-1.2, and so on. In the case of CA-1, the types “Providing too soon / too late / in wrong order” and “Stopping too soon / applying too long” are not considered to cause hazards. “Providing in wrong order” is assumed to be covered already in UCA-1.1. Similarly, “Providing too late” is assumed to be covered in UCA-1.3.

It is possible that while defining UCAs you discover hazards, which have not been previously identified in step 1. In this case, the STPA team must decide whether to take advantage of STPAs iterative nature by adding this newly identified hazard to step 1 as a new hazard or as a sub-hazard, or if labelling it in this section is sufficient. Most important is, that you make a note about your discovery, so you remember to get back to it also further into the analysis.

Please note that while the handbook suggests turning UCAs into controller constraints, we skip this step at this stage. We believe that creating separate constraints and requirements specifically for controllers is not crucial for proceeding to step 4, considering the extra effort it requires compared to the benefits it brings. However, we do recognize the importance of using STPA's findings to define constraints and requirements and apply the modification suggested by [2]. Therefore, in our guide, we perform this step at the end of step 4 when crafting safety requirements to prevent or handle loss scenarios.

4.4 Step 4: Identify Loss Scenarios

Until now, STPA focused solely on CAs and how they can become UCAs (downwards arrows). But also, feedback, other inputs, interactions (upwards and horizontal arrows), and controllers themselves can trigger UCAs, and ultimately give rise to losses. Step 4 addresses this gap and closes the (control) loop by examining all system elements for their potential to induce UCAs and cause loss (Figure 13). This will result in a list of so-called Loss scenarios. Loss scenarios describe how various causal factors in different situations combine to create hazards. [1]

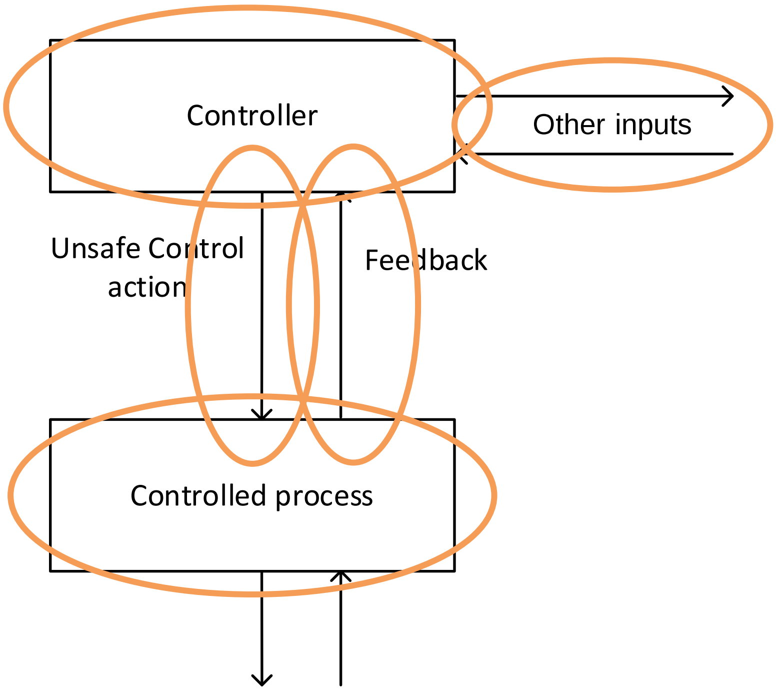


Figure 13. System elements and interactions that are analysed in STPA Step 4.

The handbook defines two types of Loss Scenarios: a) Loss Scenarios answering to the question “Why would UCAs occur?” and b) Loss Scenarios answering to the question “Why would CAs be improperly executed or not executed, leading to hazards?” Based on our experience with STPA, we recommend an approach, that directly addresses each system element with a suitable set of questions (Table 6). This simplifies and breaks down the b) type question introduced by the handbook.

For identifying loss scenarios related to each system elements proceed like this:

- Pick a system element of your choice. We recommend continuing with UCAs (UCA-1.1, UCA-1.2, etc.), as you have just previously in step 3 explored how CAs can lead to UCAs.
- Begin by brainstorming the questions suggested in Table 6 to uncover loss scenarios related to the type of system element you picked. Do not let yourself get confused with the number of questions. The purpose is to get your mind thinking about the different ways each system element can cause loss scenarios.
- Use the answers to the questions to formulate the description of the loss scenario. It is possible to have multiple loss scenarios for the same system element.
- Proceed to analyse a system element that is within the same control loop as the one you just have examined. This sequential approach helps maintain focus. Continue generating loss scenarios by asking the respective questions presented in Table 6.

Table 6. Questions to generate loss scenarios.

Loss scenarios related to:	Questions which help identify how this system element can cause a loss scenario
UCA	- Why would this UCA occur? - What needs to happen that this UCA becomes true? - Why would a controller “provide”, “not provide”, “provide too soon, too late or in the wrong order”, or “stop the CA too soon or apply it too long”? - Which factors in the controller lead to the UCA? - Which factors lead to inadequate flow of information and feedback to the controller and consequently lead to the UCA?

Feedback	- What needs to happen, that this feedback causes hazards? - Which failures in the sensor can lead to improper feedback? - How is it possible, that improper / no feedback is provided? - How can this feedback lead to an improper / no implementation of the CA?
Controller	- What needs to happen that this controller causes hazards? - How could the controller believe to be in a certain state (process model) that is actually not true? - How could the controller's process model cause a hazard? - What flaws in the control algorithm can cause the controller to "provide", "not provide", "provide too soon, too late or in the wrong order", or "stop the CA too soon or apply it too long"? - How could the controller effect the actuator to act unsafe? - What problems in the actuator can lead to hazard?
External input	- What needs to happen, that this input causes hazards? - How is it possible that improper or no inputs are provided? - How can this input lead to an improper / no implementation of the CA?

STPA practitioners are encouraged to decide themselves to which extend they want to follow the formal instructions by the Handbook. Regardless of the chosen approach, both methods can benefit from the checklist introduced in Appendix [1]. This checklist was developed based on the formal approach in the Handbook and provides a list of possible factors (a1, a2, b1, b2), which could lead to loss scenarios. Revising this checklist can greatly aid your understanding of how loss scenarios may arise and be valuable in guiding your analysis process.

Note, the STPA Handbook requires an intermediate step before the actual identification of loss scenarios can begin. Which is that actuators and sensors must be identified and modelled into the control structure of step 2. After conducting several STPAs, we have reached the conclusion to leave this sub step out of this guide. We find it impractical to make such drastic changes to the control structure at this stage. Nonetheless, we acknowledge the significance of examining how actuators and sensors may contribute to loss scenarios. We believe this can be achieved without a redesign of the control structure, or, if necessary, by fine-tuning the control structure to the desired level during step 2.

Example: Identify Loss Scenarios

We have created loss scenarios only for a selection of feedbacks, external inputs, and controllers and the UCAs identified in step 3, to prevent this example from getting too excessive. The result is displayed in Table 1. In general, you should create loss scenarios for all system elements relevant to your use case.

Table 7. Loss Scenarios for exemplary system elements.

Loss Scenario description	**Related UCA / System element**
Failure in the traffic and mission control system itself Failure in the connection of the traffic and mission control system and the AM Traffic and mission control system receives incorrect data	UCA-1.1
Failure in the traffic and mission control system itself Failure in the connection of the traffic and mission control system and the AM Traffic and mission control system receives incorrect data	UCA-1.2

Failure in the traffic and mission control system itself Failure in the connection of the traffic and mission control system and the AM Traffic and mission control system does not receive any information about entities	UCA-1.3
Failure in the connection of the traffic and mission control system and the entity Failure in the tag of the entity Entity has different dimensions than assumed by Traffic and mission control system Traffic and mission control system does not consider environmental parameters affecting the length of brake path	UCA-1.4
Failure in the tag sensors Tag is not worn by person Tag is not attached to AM Tag battery is empty and does not send signal anymore	Tag data (External input)
Failure in the traffic mission and control system itself Traffic and mission control system fails in combing drive command with environmental condition Failure in the connection of the traffic and mission control system and the AM Traffic and mission control system does receive inaccurate information about environmental condition	UCA-2.1
The parameters to define extraordinary environmental conditions are set too low Failure in the traffic mission and control system itself Failure in the connection of the traffic and mission control system and the AM	UCA-2.2
The parameters to define extraordinary environmental conditions are set too low The parameters to define normal environmental conditions are set too high Failure in the traffic mission and control system itself Failure in the connection of the traffic and mission control system and the AM	UCA-2.3
Failure in the AM sensors Failure in machine automation decision making The parameters of Status / Diagnostic Data are set too low	Status / Diagnostic Data (Feedback)
Failure in AM's sensors Failure in AM's connection to traffic and control system AM location parameters are not set accurately	Machine location (Feedback)
Is not aware what consequences his command to stop providing mission plans has, when operation is already ongoing	Remote operator (human) (Controller)
Traffic and mission control system itself receives no data Traffic and mission control system itself receives wrong data Traffic and mission control system itself receives data too late Failure in connection between traffic and mission control system and system supervisor (human) Traffic and mission control system receives correct data but fails in interpreting the data	System status (Feedback)
Failure in processing the data which the Traffic and mission control system receives and what it transmits Traffic and mission control system suffers a data overload and freezes Traffic and mission control system receives correct data but fails in interpreting the data Traffic and mission control system fails in combing drive command with environmental condition	Traffic and mission control system (Controller)

Failure in the infra tag reader itself Failure in the connection of Other traffic and items and infra tag reader Failure in the connection of the infra tag reader and traffic and mission control system Traffic and mission control system does not receive any information about entities	Locations of tagged entities withing range (External input)
Failure in the infra tag reader Parameters in infra tag reader are set too high / low Infra tag reader is not installed at optimal location Infra rag reader battery is empty and does not receive / send signal anymore	Infra tag reader (Controller)
Failure in the tag, that other traffic and items carry Failure in the connection of the tag that other traffic and items carry and infra tag reader Failure in labeling the entities (AM and person tag have been swapped and infra tag reader believes item is a AM when it is a human and vice versa)	Tag data (Feedback)

According to the handbook STPA officially ends with the definition of loss scenarios. The Handbook emphasizes the importance of defining constraints and requirements in step 1 and step 3. As we did not address this sub step in step 3, we effectively move it to this section. Because, as already mentioned, we believe that it is important to utilize especially the loss scenarios for defining operational and safety requirements and, consequently, for integrating safety into the system architecture. Also [2] recommends creating safety requirements as final step to prevent and manage loss scenarios.

Defining constraints and requirements derived from loss scenarios can be as straightforward as the substep "Defining System-Level Safety Constraints" in step 1. By engaging experts and system design engineers, accurate solutions can be readily identified. Their expertise and insights are invaluable in formulating effective measures to address the identified hazards and prevent potential loss scenarios. By this collaborative effort, the process of defining requirements is informed, and ensures that safety is integrated seamlessly into the system design.

Example: Utilizing the Loss Scenarios to identify Safety Requirements

Table 8 addresses a selection of loss scenarios. As our team comprises non-technical experts and considering that our use case is imaginary, the system requirements presented below may not appear detailed and lack specific technical specifications. They serve as a starting point to illustrate the safety measures and design considerations necessary for mitigating potential hazards in the given context.

Table 8. Safety requirements for a selection of loss scenarios.

Loss Scenarios	**Safety requirements**
Traffic and mission control system does not receive any information about entities.	Traffic and mission control system may not cause a hazard even if it does not receive information about entities. Other safety measures must take over if an object is in the way of AM.
Failure in the tag of the entity.	The tag must function all times and must not be affected by weather. It must be maintained regularly.
Entity has different dimensions than assumed by Traffic and mission control system.	Traffic and mission control system must have knowledge about all possible dimensions of entities and if in doubt must assume the biggest parameters.

Traffic and mission control system does not consider environmental parameters affecting the length of brake path.	Traffic and mission control system may not operate without considering environmental parameters.
Failure in the tag sensors.	The tag must be maintained regularly.
Tag is not worn by person.	People must be remined to wear their personal tags. A gate could ensure that only people with tags on them can enter the operation area.
Tag is not attached to AM.	AMs must be checked if they have a tag before they leave to a mission.
Traffic and mission control system does receive inaccurate information about environmental condition.	Traffic and mission control system must check if the environmental parameters are reasonable. The sensors which provide data on environmental conditions must be maintained regularly. System supervisor must regularly check upon the suitability of the environmental parameters
Failure in AM's automation decision making.	The AM's control algorithm must be checked precisely and for all possible scenarios.
The parameters of Status / Diagnostic Data are set too low.	The parameters must be adjusted to cause the correct reactions.
Remote operator is not aware what consequences his command to stop providing mission plans has, when operation is already ongoing.	Remote operator must receive proper training.

Below (Table 9) an attempt to increase understaning of why each STPA step has it’s justification and their interconnection. The table represents an example of an STPA result backwards. It begins at the final output, the loss scnario, and results at the system-level loss which is to be prevented.

Table 9. Backwards representation of STPA results.

STPA Step	Output of the step	Description of the output
4	Loss Scenario	Traffic and mission control system fails in combing drive command with operational context.
3	UCA	Traffic and mission control system provides drive command to route point, but the command is not adjusted to the given operational context.
2	CA	Drive command to route point
1	SC	The movement of the autonomous machine shall not cause risk of collision when there are vehicles or other objects in the way
1	System-Level hazard	The AM moves when there are vehicles or other objects in the way.
1	System-Level Loss	Loss of, or damage to vehicle (AM or others).

It is quite common, that your STPA results in an overwhelming list of loss scenarios. Prioritization could bring clarity and help identify which loss scenarios should be prevented urgently from happening. One possibility is to apply the Risk Priority Number approach [12], which chapter 5.1 introduces.

5. Extensions

This section serves as brief overview on what literature offers to enhance performance and results of STPA. The presented STPA extensions are optional improvement suggestions.

5.1 Prioritizing STPA Results

Despite yielding numerous UCAs and Loss scenarios—sometimes reaching hundreds or thousands based on control structure detail—STPA lacks an inherent prioritization procedure. This lack of differentiation may cause an unnecessary workload for those tasked with translating STPA findings into system and safety requirements. All UCAs and loss scenarios appear equally important, even though some findings must be treated with more care in terms of criticality for safety requirements and system design considerations.

Integrating the Risk Priority Number (RPN) approach [12] can streamline the workload by identifying and eliminating less critical UCAs and Loss scenarios already during the analysis. RPN-based prioritization allows screening out less relevant UCAs during STPA step 3, thereby reducing the originally required efforts in step 4. This approach enables resources to focus on creating and prioritizing the remaining Loss scenarios.

Figure 14 below shows the integration of the RPN approach into STPA step 3 (Figure 14, 3-2 and 3-3) for screening UCAs, and its subsequent use to prioritize Loss scenarios in step 4 (Figure 14, 4-2 and 4-3).

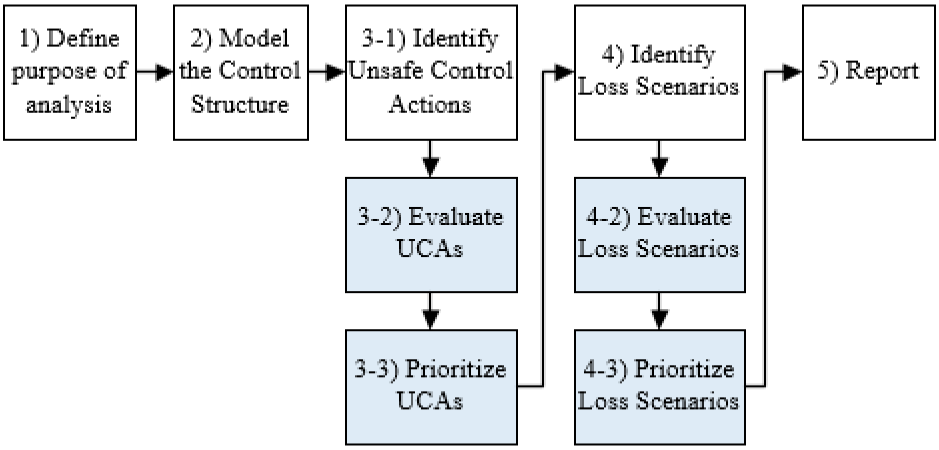

1) Define purpose of analysis
2) Model the Control Structure
3-1) Identify Unsafe Control Actions
4) Identify Loss Scenarios
5) Report
3-2) Evaluate UCAs
4-2) Evaluate Loss Scenarios
3-3) Prioritize UCAs
4-3) Prioritize Loss Scenarios

Figure 14. RPN approach integrated into STPA procedure. (modified from [12]*)*

To identify which UCAs can be neglected, multiply estimates for severity (SV), available time to respond (ATR), and strength of knowledge on UCA (SOK_{UCA}). The resulting value is called RPN_{UCA} and is obtained through the formula:

$$RPN_{UCA} = SV \times ATR \times SOK_{UCA}$$

Once you have all RPN_{UCA} values, focus your efforts to derive Loss scenarios from critical UCAs. Loss scenarios are then prioritized by calculating their individual RPN ($RPN_{LossScenario}$). The estimation criteria for $RPN_{LossScenario}$ are likelihood (LH) and strength of knowledge on Loss scenario ($SOK_{LossScenario}$) and are multiplied with the corresponding RPN_{UCA} as represented in the following formula:

$$RPN_{LossScenario}= RPN_{UCA} \times LH \times SOK_{LossScenario}$$

Each criterion's estimation ranges from 1 to 5, with lower numbers indicating less severe damage (SV), a slow transition time between UCA resulting in loss (ATR), and strong overall knowledge of the analyst on UCA (SOK_{UCA}), respectively on loss scenario ($SOK_{LossScenario}$) and the event to happen rarely (LH). By estimating the STPA practitioner’s strength of knowledge on UCA or Loss scenario, the RPN reflects the level of confidence or certainty in predicting the occurrence and outcome of a UCA or a Loss scenario [12].

We argue, that in order to perform cuts at the appropriate places (remove UCAs from analysing their potential to cause loss scenarios) system experts must be deeply involved throughout the process. Further, we noticed that screening UCAs and prioritizing Loss scenarios shifts the workload towards estimating suitable criteria rather than reducing it. However, the benefits of prioritization become apparent when using STPA results to impact system safety constraints, formulate safety goals, or influence system design.

VTT will publish a deliverable within the “Systems Engineering approaches for managing the life cycle of I&C systems” project (SEAMLES), which sheds more light on how to apply this approach.

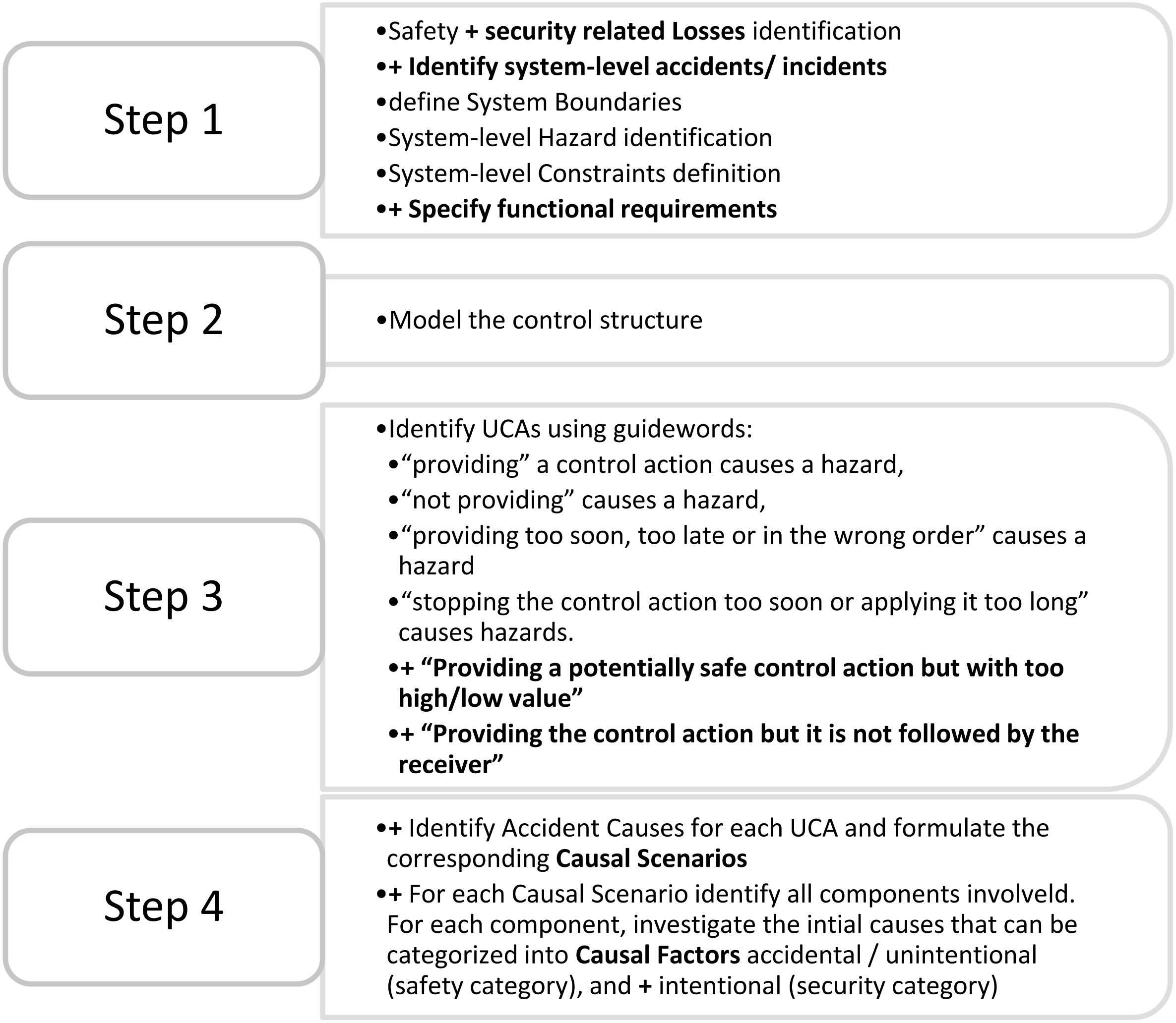
5.2 Extension for Safety and Security co-analysis

In recent years, there has been a growing need for safety and security co-analysis. This is because modern safety-critical systems are highly interconnected, and they cannot be considered safe unless they are also secure. By conducting a unified safety and security analysis, one can not only achieve the goal of creating systems that are both safe and secure but also gain an understanding of how these two aspects interrelate, meaning, the impact of security on safety, as well as safety on security. [19]

In 2013, Young and Leveson [20] introduced the first extension of the basic STPA, which also incorporates security analysis, known as STPA-Sec. Since then, several approaches to combine safety and security analysis by either utilizing STAMP, STPA or STPA-Sec have emerged. Schmittner [21] proposed enhancements to STPA-Sec, primarily involving minor adjustments. Temple [22] combined STPA-Sec with Failure Mode, Vulnerabilities and Effect Analysis (FMVEA) to establish the systems-theoretic likelihood and severity analysis (STLSA). Shapiro [23] introduced STPA-Priv by adapting STPA-Sec to assess system controls for their ability to compromise privacy. In contrast, some researchers have made substantial modifications to the original framework. For instance, Friedberg [24] developed STPA-SafeSec. Further, studies [25], [26] have combined STPA with STRIDE to analyse for cyber security threats. Moreover, Pereira [27] and Troubitsyna [28] have integrated STPA with NIST SP800-30 and Goal Structuring Notation, respectively, to create their own safety and security co-analyse methods.

In this guide, we draw the focus to the so called “STPA-Extension” method ([29], [30]) which facilitates the joint analysis of safety and security issues. We consider this method particularly advantageous for STPA beginners, as it can be implemented with minimal modifications to the fundamental steps of STPA. The STPA-Extension focuses rather on security vulnerabilities of the operational technology than IT and therefore is considered more useful for manufacturers of autonomous vehicles and systems in which they operate. Since the generic STPA is already a complex and labour-intensive process we recognize the value of providing practitioners with the choice to freely opt for this additional security extension if needed.

Figure 15 gives an overview of the steps that the STPA Extension adds (labelled with a “+” symbol) to the generic STPA.



Step 1

- Safety **+ security related Losses** identification
- **+ Identify system-level accidents/ incidents**
- define System Boundaries
- System-level Hazard identification
- System-level Constraints definition
- **+ Specify functional requirements**

Step 2

- Model the control structure

Step 3

- Identify UCAs using guidewords:
  - “providing” a control action causes a hazard,
  - “not providing” causes a hazard,
  - “providing too soon, too late or in the wrong order” causes a hazard
  - “stopping the control action too soon or applying it too long” causes hazards.
  - **+ “Providing a potentially safe control action but with too high/low value”**
  - **+ “Providing the control action but it is not followed by the receiver”**

Step 4

- **+** Identify Accident Causes for each UCA and formulate the corresponding **Causal Scenarios**
- **+** For each Causal Scenario identify all components involveld. For each component, investigate the intial causes that can be categorized into **Causal Factors** accidental / unintentional (safety category), and **+** intentional (security category)

Figure 15. Additional steps of the STPA Extension analysis process. Adapted from [29]

Just like the generic STPA, the STPA Extension also concludes with the definition of loss scenarios. However, in this extension, they are referred to as Causal Scenarios. Causal Scenarios describe the effects that an UCA could lead to. These effects are determined by applying a set of accident causes to each UCA and evaluating their relevance. The accident causes used to identify Causal Scenarios are [27]:

AC1. Control input or external information wrong or missing
AC2. Inadequate control algorithm
AC3. Process model or feedback wrong, missing or inadequate
AC4. Inadequate CA actuation

Next, every component involved in the Causal Scenario is analysed for potential failure modes. The final step identifies the underlying causal factors behind these failure modes. This means that each system element involved is examined for the failures, vulnerabilities, and deliberate actions that could lead to the specific failure mode. Additionally, causal factors are categorized as either accidental or intentional.

While this guide lists various options for performing a co-analysis with STPA, detailed instructions will be provided in future work. Currently, we are conducting a safety and security co-analysis using the STPA Extension in the ongoing Business Finland funded “Connected Mobile Machine Lifetime Cyber Security” project. Further information can be retrieved from https://www.six.fi/comma.

5.3 Extension for Human Machine Interaction

Systems may contain one or more human elements, for example in the role of a controller, operator, or supervisor. Humans represent the most complex type of system components, thus exploring this element further opens up a whole new range of considerations. This is because human-machine interactions (HMI) are influenced by humans' previous learnings, perceptions, and beliefs, which, in turn, affect commands, actions and decision-making processes. Additionally, organizational culture shapes human behaviour, either enhancing the human potential to act safely and prevent accidents or inducing the opposite. Even the design of the operational environment can impact humans, for example, by increasing the possibility of confusing operational modes and losing situational awareness, which can lead to hazards. Therefore, it is important to be aware that human behaviour cannot be automated and can be unexpected and possibly introduces unique vulnerabilities and challenges to the system [1]. While the handbook [1] provides little instructions for understanding why humans behave in a certain way and how to systematically analyse safety impacts due to human perception and action, we aim to close this gap. Therefore, in this chapter, we provide guidance to STPA practitioners on HMI extension introduced by France [9].

The STPA extension for HMI aids in identifying combinations of circumstances and assumptions that can cause human induced loss scenarios. It provides the necessary steps to successfully identify safety hazards originating from humans' flawed mental models and decision-making. Similar to non-human system elements, this extension helps conclude system requirements, necessary behaviour constraints, and specifications for educational trainings to design and adjust the system to promote, prevent, and partially steer human behaviour [1], [2], [9].

In Figure 8 we already used different colours to highlight the human elements in the control structure. However, if the scope of the STPA stresses the HMI, we suggest adjusting the control structure by zooming into the human factor extension with more detail. Therefore, define elements of the human factor extension, like Mental Model Updates (U), Process State (PS), Mental Models about Process Behaviour (PB), Environment (E) and Control Action Selection (S).[9] [2] Please view the definitions introduced in Table 10. Figure 16 illustrates these elements and how their interactions form the human controller's decision on CAs.

Table 10. Definitions of terms for the human factor extension. Adapted from [2]

Term	Definition
Control Action Selection	– Captures the human's goals and how decisions are made based on mental models
Mental Model of Process State	– Human's believes about modes, current process stage, system variables, etc.
Mental Model of Process Behaviour	– Human's believes about what the system can do, how the system will behave in a particular mode, if-then relationships between driver and system output
Mental Model of Environment	– Changes in environmental conditions / operation environment, familiar or unfamiliar environment, state, and behaviour of other system elements
Mental Model Updates	– Captures the influence of human experiences, and expectations on the processing of the received input

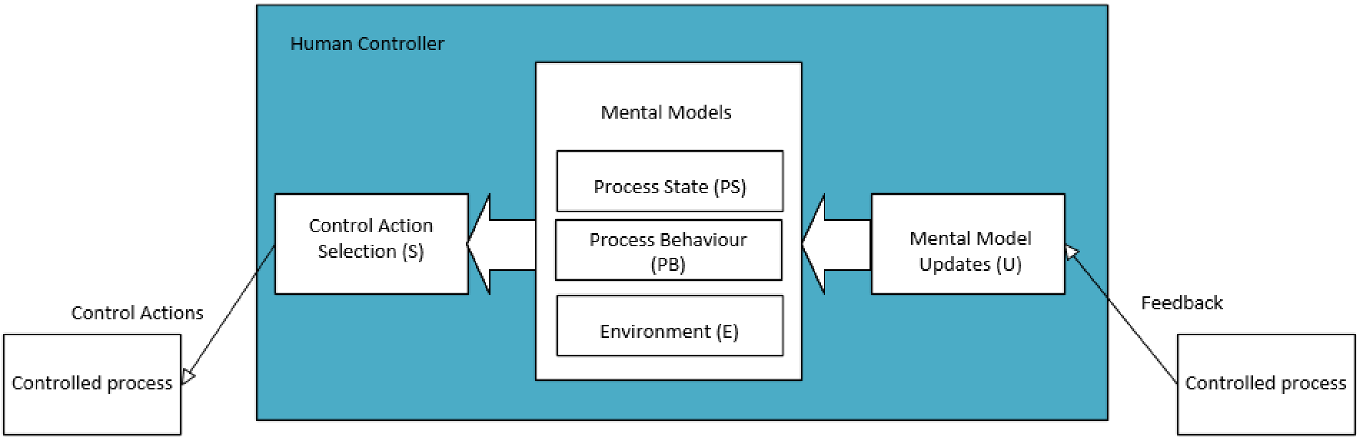


Figure 16. Human controller model within the control loop. Adapted from [9]

STPA practitioners can identify the human factor extension elements by following lead questions [9]:

- S: How did the operator choose which CA to perform?
- Mental Models about PS, PB, E: What does the operator know or believe about the system?
- U: How did the operator come to have their current knowledge or beliefs?

Apply these lead questions and their corresponding subquestions gathered in

Table 11. They guide you to identify UCAs and loss scenarios related to human factor. Also, feel free to customize these questions to suit your analysis and to suit the role the human may have within your system.

Table 11. Questions to identify human factor extension elements. Adapted from [2]

Human factor extension element	Questions to support identification
Control Action Selection	**How does the human decide what CA to perform?** What are the human's goals? What alternatives is the human choosing between? How automatic or novel is the behaviour? How might the human's mental models affect their decisions? What external factors (e.g. time pressure) might affect their decisions?
Mental Model of Process State	**What are the human's beliefs about**… - modes and mode changes? - the current process stage (if multiple?) - system variables (e.g. true/false, on/off)?
Mental Model of Process Behaviour	**What are the human's beliefs about**… - what the system can do? - how will the system behave in a particular mode or stage of operation? - if-then relationships between human's input and system output?
Mental Model of Environment	**What are the human's beliefs about**… - state and behaviour of *other* system elements (also humans) in the environment? - social and organizational relationships in the environment? - changes to the environment? - familiarity of the environment?
Mental Model Updates	**How did the human come to have their current beliefs?** How were the human's beliefs and mental models formed initially? What might have triggered (or not triggered) an update to these mental models? Are there non-feedback inputs such as training programs and documentation? Was input/feedback actually observed, or was it missed due to low salience or expectations? Was input/feedback correctly interpreted, or was it misunderstood?

6. Conclusion

We hope that this guide will be a valuable resource for both current and future STPA practitioners as you endeavour to assess the safety of complex systems. Our aim is not only to provide assistance but also to underscore the advantages inherent in employing this safety analysis method.

With the aid of STPA, you can systematically analyse intricate systems without becoming overwhelmed by their complexity. This method allows you to break down the analysis into manageable steps, providing the opportunity to scrutinize each CA and feedback individually, giving each the attention it deserves. From our experience, UCAs and Loss scenarios identified through STPA prove to be invaluable inputs for system design and defining safety requirements. It is essential to recognize that the control structure plays a pivotal role in successful STPA, as a substantial part of the analysis relies on its content. Careful consideration is needed when deciding to omit specific CAs, as neglecting them may result in overlooked UCAs and subsequent Loss scenarios. We advise relying on the estimation of your system experts to ensure the hierarchical control structure accurately represents the use case.

Understanding the theory of STPA may initially seem exhaustive, even with the aid of this guide; however, we assure you that conducting an STPA is not as complex as it may appear. For those new to STPA, seeking support from experienced practitioners is encouraged. Nevertheless, learning STPA independently is feasible, and this guide aims to make the process more accessible. Begin with a simple use case and focus on the essential four steps. Keep in mind that documenting UCAs and Loss scenarios involves repetitions and cross-references, so do not hesitate to utilize the traceability feature to make adjustments and refinements.

While our documentation of STPA in MS Excel and MS Visio has been practical, the limitations of these platforms hindered effective traceability. We encourage you to try the suitability of available software tools and experience if it increases focus on analysis.

We wish you the best of luck on your learning journey and in conducting fruitful STPA analyses.

References

[1] N. Leveson and J. Thomas, *STPA Handbook*. 2018.

[2] “SAE J3187 - System Theoretic Process Analysis (STPA) Recommended Practices for Evaluations of Automotive Related Safety-Critical Systems,” *SAE International*, Art. no. J31875, 2022, doi: https://doi.org/10.4271/J3187_202202.

[3] E. Heikkilä, T. Malm, J. Sarsama, R. Tiusanen, and T. Ahonen, “Hazard Analysis of an Autonomous Container Handling System – a Comparison of STPA and HAZOP Methods,” *Scientific Journal of Gdynia Maritime University*, no. 125, pp. 25–39, Mar. 2023, doi: 10.26408/125.02.

[4] Y. Zhang, C. Dong, W. Guo, J. Dai, and Z. Zhao, “Systems theoretic accident model and process (STAMP): A literature review,” *Safety Science*, vol. 152. Elsevier B.V., Aug. 01, 2022. doi: 10.1016/j.ssci.2021.105596.

[5] J. Thomas and W. Young, “STPA Standards, Certification, and Accreditation.” 2023. Accessed: Dec. 11, 2023. [Online]. Available: http://psas.scripts.mit.edu/home/wp-content/uploads/2023/2023-06-08-1140__John-Thomas__PUB.pdf

[6] J. Alanen, J. Linnosmaa, J. Pärssinen, A. Kotelba, and E. Heikkilä, “Review of cybersecurity risk analysis methods and tools for safety critical industrial control systems,” 2022.

[7] R. Tiusanen, E. Heikkilä, and T. Malm, “System Safety Engineering Approach for Autonomous Mobile Machinery,” in *Lecture Notes in Mechanical Engineering*, Springer Science and Business Media Deutschland GmbH, 2021, pp. 239–251. doi: 10.1007/978-3-030-64228-0_21.

[8] N. Leveson, *Engineering a Safer World*. Massachusetts Institute of Technology, 2011.

[9] M. E. France, “Engineering for Humans: A New Extension to STPA,” Master’s Thesis, Massachusetts Institute of Technology, 2017. Accessed: Dec. 04, 2023. [Online]. Available: http://hdl.handle.net/1721.1/112357

[10] N. Leveson, “A new accident model for engineering safer systems,” 2004, doi: 10.1016/S0925-7535(03)00047-X.

[11] N. A. Zikrullah and H. Kim, “Prioritization Approach for Systems-Theoretic Process Analysis (PA-STPA) : Applied for Subsea Systems,” Norwegian University of Science and Technology, Master’s Thesis, 2018. Accessed: Dec. 04, 2023. [Online]. Available: http://hdl.handle.net/11250/2562563

[12] H. Kim, M. A. Lundteigen, A. Hafver, and F. B. Pedersen, “Utilization of risk priority number to systems-theoretic process analysis: A practical solution to manage a large number of unsafe control actions and loss scenarios,” *Proc Inst Mech Eng O J Risk Reliab*, vol. 235, no. 1, pp. 92–107, Feb. 2021, doi: 10.1177/1748006X20939717.

[13] ISO, “ISO 21448:2019 - Road vehicles - Safety of the intended functionality,” 2019.

[14] N. A. Zikrullah, “Contributions to the safety of novel subsea technologies - Methods and approaches to support the safety demonstration process,” Norwegian University of Science and Technology, 2022.

[15] S. V. Blindheim, “Risk-aware decision-making and control of autonomous ships,” Norwegian University of Science and Technology, 2023.

[16] N. A. Zikrullah, M. J. P. Van Der Meulen, G. Skofteland, and M. A. Lundteigen, “A Comparison of Hazardous Scenarios in Architectures with Different Integration Types,” in *Proceedings of the 30th European Safety and Reliability Conference and the 15th Probabilistic Safety Assessment and Management Conference*, 2020. doi: 10.3850/978-981-14-8593-0.

[17] J. Petzold, “A Textual Domain Specific Language for System-Theoretic Process Analysis,” Kiel University, 2022.

[18] S. Blindheim, · Tor, A. Johansen, and I. B. Utne, “Risk-based supervisory control for autonomous ship navigation,” *J Mar Sci Technol*, doi: 10.1007/s00773-023-00945-6.

[19] E. Lisova, I. Šljivo, and A. Čaušević, “Safety and Security Co-Analyses: A Systematic Literature Review,” *IEEE Systems Journal*, vol. 13, no. 3. Institute of Electrical and Electronics Engineers Inc., pp. 2189–2200, Sep. 01, 2019. doi: 10.1109/JSYST.2018.2881017.

[20] W. Young and N. Leveson, “Systems thinking for safety and security,” in *ACM International Conference Proceeding Series*, 2013, pp. 1–8. doi: 10.1145/2523649.2530277.

[21] C. Schmittner, Z. Ma, and P. Puschner, “Limitation and Improvement of STPA-Sec for Safety and Security Co-analysis,” in *Computer Safety, Reliability, and Security*, in Lecture Notes in Computer Science, vol. 9923. 2016. doi: 10.1007/978-3-319-45480-1.

[22] W. G. Temple, Y. Wu, B. Chen, and Z. Kalbarczyk, “Systems-Theoretic Likelihood and Severity Analysis for Safety and Security Co-engineering,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Springer Verlag, 2017, pp. 51–67. doi: 10.1007/978-3-319-68499-4_4.

[23] S. Shapiro, “Privacy Risk Analysis Based on System Control Structures Adapting System-Theoretic Process Analysis for Privacy Engineering,” in *IEEE Security and Privacy Workshops (SPW)* , 2016, pp. 17–24. doi: 10.1109/SPW.2016.15.

[24] I. Friedberg, K. McLaughlin, P. Smith, D. Laverty, and S. Sezer, “STPA-SafeSec: Safety and security analysis for cyber-physical systems,” *Journal of Information Security and Applications*, vol. 34, pp. 183–196, Jun. 2017, doi: 10.1016/j.jisa.2016.05.008.

[25] T. Kaneko, R. Sasaki, and Y. Takahashi, “Threat analysis using STRIDE with STAMP/STPA,” 2019.

[26] N. P. de Souza, C. de A. C. César, J. de M. Bezerra, and C. M. Hirata, “Extending STPA with STRIDE to identify cybersecurity loss scenarios,” *Journal of Information Security and Applications*, vol. 55, Dec. 2020, doi: 10.1016/j.jisa.2020.102620.

[27] D. Pereira, C. Hirata, R. Pagliares, and S. Nadjm-Tehrani, “Towards combined safety and security constraints analysis,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Springer Verlag, 2017, pp. 70–80. doi: 10.1007/978-3-319-66284-8_7.

[28] E. Troubitsyna, “An Integrated Approach to Deriving Safety and Security Requirements from Safety Cases,” in *Proceedings - International Computer Software and Applications Conference*, IEEE Computer Society, Aug. 2016, pp. 614–615. doi: 10.1109/COMPSAC.2016.58.

[29] N. H. Carreras Guzman, J. Zhang, J. Xie, and J. A. Glomsrud, “A Comparative Study of STPA-Extension and the UFoI-E Method for Safety and Security Co-analysis,” *Reliab Eng Syst Saf*, vol. 211, p. 107633, Jul. 2021, doi: 10.1016/j.ress.2021.107633.

[30] J. A. Glomsrud and J. Xie, “A Structured STPA Safety and Security Co-analysis Framework for Autonomous Ships,” 2019, doi: 10.3850/978-981-11-2724-3.

Appendix

The handbook defines two types of Loss Scenarios: a) Loss Scenarios answering to the question "Why would UCAs occur?" and b) Loss Scenarios answering to the question "Why would CAs be improperly executed or not executed, leading to hazards?"

Loss Scenarios can be created based on the selection of possible factors (a1, a2, b1, b2) leading to a hazardous situation. Loss Scenarios are written descriptions of causal factors in different situational contexts aiming to grasp how combinations of factors can lead to hazard.

a) identifies scenarios that lead to UCAs. It starts with the UCA itself and then identifies backwards why a controller would "provide", "not provide" etc. a CA.

UCAs can be caused by **unsafe controller behaviour** (a1) and by **inadequate feedback and other inputs** (a2). The following checklist provides examples on each of these factors.

a1. Causes of unsafe controller behaviour

- Failure of the controller
  - Physical failure
  - Power failure
- Inadequate control algorithm
  - Flawed implementation of the specified control algorithm
  - The specified control algorithm is flawed
  - The specified control algorithm becomes inadequate over time due to changes or degradation
  - Security: Control algorithm flaw is introduced by an adversary
- Unsafe control inputs (usually identified in the previous step)
  - UCA received from another controller
- Inadequate process models:
  - Controller receives incorrect feedback/information
  - Controller receives correct feedback/information but interprets it incorrectly or ignores it
  - Controller does not receive feedback/information when needed (delayed or never received)
  - Necessary controller feedback/information does not exist

a2. Causes of inadequate feedback and information

- Feedback or information not received:
  - Feedback/info sent by sensor(s) but not received by controller
  - Feedback/info is not sent by sensor(s) but is received or applied to sensor(s)
  - Feedback/info is not received or applied to sensor(s)
  - Feedback/info does not exist in control structure or sensor(s) do not exist
- Inadequate feedback is received:
  - Sensor(s) respond adequately but controller receives inadequate feedback/info
  - Sensor(s) respond inadequately to feedback/info that is received or applied to sensor(s)
  - Sensor(s) are not capable or not designed to provide necessary feedback/info
  - Security: Specified feedback and other information is affected by an adversary

b) assumes that scenarios can also be caused without UCAs being involved. For example, if control actions are improperly or not executed. Therefore, this type investigates a wider range of underlaying problems leading to improperly or no execution of control actions.

To identify this type of Loss Scenario factors that affect the **control path** and factors that affect the **controlled process** must be considered. The control path includes control actions that are sent from the controller to the actuator who then impacts the controlled process. Factors that affect the controlled process refer to horizontal arrows in the control structure. They may consist of other controllers, and different inputs from outside.

The following checklist provides examples on scenarios involving the control path and scenarios related to the controlled process.

b1. Scenarios involving the control path

- Control action not executed
  - Control action is sent by controller but not received by actuator(s)
  - Control action is received by actuator(s) but actuator(s) do not respond
  - Actuator(s) responds but the control action is not applied to or received by the controlled process
- Control action improperly executed
  - Control action is sent by controller but received improperly by actuator(s)
  - Control action is received correctly by actuator(s) but actuator(s) respond inadequately
  - Actuator(s) respond adequately, but the control action is applied or received improperly at the controlled process
  - Control action is not sent by controller, but actuators or other elements respond as if it had been sent

b2. Scenarios related to the controlled process

- Control action not executed
  - Control action is applied or received by the controlled process but the controlled process does not respond
- Control action improperly executed
  - Control action is applied or received by the controlled process but the controlled process responds improperly
  - Control action is not applied or received by the controlled process but the process responds as if the control action had been applied or received [1]

DocuSign

Certificate Of Completion

Envelope Id: B6456698643C4E8091CA4B6E050073C4		Status: Completed
Subject: Complete with DocuSign: 2024-1-2_STPA guide_Final Version.pdf		
Source Envelope:		
Document Pages: 43	Signatures: 1	Envelope Originator:
Certificate Pages: 1	Initials: 0	Jessica Vepsäläinen
AutoNav: Enabled		Tekniikantie 21, Espoo
EnvelopeId Stamping: Enabled		., . P.O Box1000, FI-0204
Time Zone: (UTC+02:00) Helsinki, Kyiv, Riga, Sofia, Tallinn, Vilnius		Jessica.Vepsalainen@vtt.fi
		IP Address: 89.27.92.100

Record Tracking

Status: Original 03 January 2024 | 08:58	Holder: Jessica Vepsäläinen Jessica.Vepsalainen@vtt.fi	Location: DocuSign

Signer Events	Signature	Timestamp
Johannes Hyrynen Johannes.Hyrynen@vtt.fi Lead, Low carbon and smart machines Teknologian tutkimuskeskus VTT Oy Security Level: Email, Account Authentication (None), Authentication	DocuSigned by: 1F65932542874A1... Signature Adoption: Uploaded Signature Image Using IP Address: 91.153.195.73	Sent: 03 January 2024 | 09:06 Viewed: 03 January 2024 | 10:50 Signed: 03 January 2024 | 10:51

Authentication Details
SMS Auth:
Transaction: 52cca7b2-215a-4306-81ac-8ff3042d06e9
Result: passed
Vendor ID: TeleSign
Type: SMSAuth
Performed: 03 January 2024 | 10:49
Phone: +358 40 8336364

Electronic Record and Signature Disclosure:
Not Offered via DocuSign

In Person Signer Events	Signature	Timestamp

Editor Delivery Events	Status	Timestamp

Agent Delivery Events	Status	Timestamp

Intermediary Delivery Events	Status	Timestamp

Certified Delivery Events	Status	Timestamp

Carbon Copy Events	Status	Timestamp

Witness Events	Signature	Timestamp

Notary Events	Signature	Timestamp

Envelope Summary Events	Status	Timestamps
Envelope Sent	Hashed/Encrypted	03 January 2024 | 09:06
Certified Delivered	Security Checked	03 January 2024 | 10:50
Signing Complete	Security Checked	03 January 2024 | 10:51
Completed	Security Checked	03 January 2024 | 10:51

Payment Events	Status	Timestamps