A GENERIC APPROACH TO ANALYSING FAILURES IN HUMAN - SYSTEM INTERACTION IN AUTONOMY

Marilia A. Ramos[1,2], Christoph A. Thieme[1], Ingrid B. Utne[1], Ali Mosleh [1,2]

Affiliations

[1]Department of Marine Technology, Norwegian University of Science and Technology, Trondheim Norway; [2] B. John Garrick Institute for the Risk Sciences, University of California Los Angeles, USA

Abstract

Autonomous systems operation will in the foreseeable future rely on the interaction between software, hardware and humans. Efficient interaction and communication between these agents are crucial for safe operation. Conventional methods for hazard identification and safety assessment focus often on one of the aspects of the system only, e.g., human reliability, software failures, or equipment reliability. The method Human-System Interaction in Autonomy (H-SIA) was recently proposed, focusing on autonomous ships operation and collision scenarios. H-SIA provides a framework for analyzing autonomous ship operation as an entirety, rather than each agent separately. The method comprised initially of two main elements: An Event Sequence Diagram (ESD) and a Concurrent Task Analysis (CoTA). While the ESD models the events that can take place following an initiating event, the CoTA models which tasks the agents must perform for these events to succeed. This paper extends H-SIA to include the paths to failure, through the development of Fault Trees (FTs), which is necessary for risk analysis and identification of risk reduction measures. The FTs development of H-SIA introduces novelties in comparison to common FTs: i) they model the system as whole, ii) they are generic and can accommodate a diversity of systems designs; iii) they lead to basic failure events. The FTs allows for identification of failure events arising through interaction between autonomous ship and human operators, as well as failure propagation through these agents. The basic failure events are applicable for different autonomous concepts. A case study on autonomous ship collision demonstrates the use of the extended method. The case study illustrates H-SIA's applicability to different designs and levels of autonomy, its potential for identification of failure events, and its use in risk assessments.

Keywords: autonomous systems, autonomous ship, risk assessment, human-autonomy interaction, system design

1. INTRODUCTION

Autonomous capabilities are increasingly applied to a variety of systems, such as autonomous cars, ships, trains, autonomous underwater vehicles and drones. The term "autonomous system" may suggest, at a first glance, a concept in which the system is fully responsible for all aspects of its

operation and is independent of humans. Autonomous systems may, however, operate at different levels of autonomy (LoAs). For instance, an autonomous system may be remotely controlled by a human operator, may operate semi-autonomously with supervision by a human, or may be highly autonomous with limited supervision and monitoring (Rødseth and Nordahl, 2017; Vagia et al., 2016; Vagia and Rødseth, 2019). Table 1 summarizes some LoA for autonomous marine vehicles, adapted from(Utne et al., 2017). The changes between LoAs are gradual and different aspects may be automated within one LoA. Furthermore, the system may operate within one LoA by design or adapt its LoA during operation based on external factors or as part of a troubleshooting mechanism. The latter is considered a dynamic or adaptive LoA.

TABLE 1 SUMMARY OF DIFFERENT LOAS EXPECTED FOR AUTONOMOUS MARINE VEHICLES, ADAPTED FROM (Utne et al., 2017)

LoA	Name	Description
1	Automatic operation/ remote control	The operators handle all high-level mission planning tasks. They give input to waypoints, or preprogramme the system. The human operator can see all information from sensors through a Human Machine interface, such as video images, system states, etc.
2	Management by consent	The system performs many functions independent of the operator, this includes planning functions. The system may be unsupervised during parts of the mission. If changes to the mission plan are necessary, the operator needs to approve these.
3	Semi-autonomous operation/ management by consent	The system is mainly taking decisions on its own and planning the mission. The operator may change mission parameters, if necessary. The system will only alert the operator in case an unexpected situation that cannot be handled is arising.
4	Highly autonomous operation	The system acts and decides intelligent and independent. The operators are only informed about mission progress but do not have the possibility to influence the mission.

Systems designed with a LoA such that they can operate without human supervision or intervention are not expected to be launched in the near or intermediate future. For instance, autonomous ships are mainly being projected to have a control center onshore from where humans can supervise and monitor the ships and take over control remotely (Ramos et al., 2019; MUNIN, 2016; Porathe et al., 2014). Several recent publications have addressed the human operator performance when operating autonomous ships, e.g., (Di et al., 2020; Hogenboom et al., 2020).

Most autonomous systems will thus rely on the interaction between software, hardware and humans for operation. This reliance is not a novelty and is already present in complex highly automated systems (Endsley, 2017; Ho et al., 2011; Kari et al., 2018). Yet, autonomy brings new layers of complexity (Huang et al., 2010). Software is responsible for tasks of information processing and decisions that used to be human responsibilities. Humans are generally displaced from active tasks to monitoring ones while still needing to be "in the loop" (Endsley, 2017; Ho et al., 2011). Hardware must be reliable and resilient to operate for a long time without human intervention, especially in cases of unmanned systems. Above all, the hardware, software and human must interact and communicate efficiently. Thus, it is of high importance to analyze not only how hardware, software and humans

interact in autonomous systems, but also the hazards that may derive from this interaction, and how failures can propagate(Leveson, 2012; Ramos et al., 2020). A recent study by (Fan et al., 2020) identifies relevant influencing factors for the different phases of operation for MASS.

Conventional methods for hazard identification and safety assessment focus on one of the aspects of the system only, e.g., human reliability, software failures, and equipment reliability (Leveson et al., 2012; Mosleh, 2014). The interaction among components and emerging complexity is often neglected or reduced to a minimum, because it is too difficult to investigate. Autonomous systems may thus not be sufficiently well analyzed by conventional methods. For instance, risk analyses of conventional ships often do not sufficiently focus on the functions carried out by software based systems and often treat the human element superficially (Thieme et al., 2018). The concept of dynamic LoA also poses a challenge for conventional risk methods. Methods such as the System Theoretic Process Analysis (STPA) (Leveson and Thomas, 2018) or the Functional Resonance Analysis Methodology (FRAM, Hollnagel, 2017) have been developed for hazard identification of complex systems. STPA views hazards and accidents as a control problem, where control actions are insufficient and may lead to an accident. FRAM is based on resilience thinking and analysis how process variability may lead to abnormal system conditions. STPA has been applied to analyze autonomous ship operation, e.g., in (Wróbel et al., 2018a, 2018b; Zou, 2018). STPA and FRAM give input to risk analysis, but they are not quantitative methods. They do not provide directly a risk model or guidance to build a semi- or quantitative risk model (Patriarca et al., 2017; Utne et al., 2020).

Considering the above, (Ramos et al., 2020) introduced the method Human-System Interaction in Autonomy (H-SIA). They present an approach to implement identified interactions between a system's agents into semi-or quantitative risk models. The method as presented in (Ramos et al., 2020) comprises an Event Sequence Diagram (ESD) and a Concurrent Task Analysis (CoTA). H-SIA provides a framework for analyzing autonomous ships operation as an entirety. The use of the same method, level of analysis and language for analyzing software, hardware and humans aims at identifying the interactions between the agents in a comprehensible and cohesive manner.

The initial method focuses on autonomous ships (AS) operation and collision scenarios, considering the autonomous system and the human operator. The ESD reflects collision scenarios with explicit inclusion of both autonomous system and human events. The CoTA is a novel method. It is a success-oriented tool, based on Task Analysis theory, which translates the events of the ESD into goals to be achieved. However, the CoTA does not provide all the necessary elements for a failure-oriented analysis. Risk analysis of an operation requires the identification of failure events, the paths to failure and its causes. In order to be used for risk analysis it is necessary, thus, to extend the H-SIA framework.

This paper extends H-SIA to include the paths to failure, through the development of Fault Trees (FTs). The FTs are connected to the ESD in a top-down approach. The joint use of ESD and FTs follow a Hybrid Causal Logic (HCL) modeling approach (Wang, 2007, Groth et al., 2010, 2009; Roelen et al., 2008). HCL is an effective approach to modeling risks associates with complex systems. It uses event sequence diagrams as the first layer for describing system behavior in case of anomalies, and then provides more detailed picture of the contributing causes by FTs (Wang, 2007). H-SIA aims at using established risk assessment methods (ESDs, Task analyses, FTs). These are enhanced with the features necessary to address the challenges autonomous systems pose: the need for a holistic analysis, considering interactions between a system's agents and the capability to reflect dynamic LoAs.

The distinguished features of the CoTA are reflected in the FT, and the outcome is a set of interconnected FTs that cover failures not only within one sub-system (e.g. human failures, software failures) but also failures arising from the interaction among the sub-systems. Several advantages are expected from applying the presented FTs. Firstly, they follow the main feature of H-SIA: they model the system as whole. The FTs are developed for the technical system (software and hardware) and the human concurrently. They are connected through gates and explicitly model the propagation of failures between the agents. Secondly, the FTs are generic and can accommodate a diversity of system designs. They can be adapted to the chosen vessel design and the context of the identified scenario. Thirdly, the FTs lead to generic basic failure events. These can be as detailed as needed for the analysis, using software, hardware or human analysis methods. The basic failure events are generic and therefore applicable for different autonomous vessel concepts. The dependencies among the failures of different sub-systems can easily be overlooked when using analysis methods that do not account for the whole system and its interactions, as the CoTA and the derived FTs do.

The extended H-SIA method is demonstrated with a case study on autonomous ship collision. The case study is, initially, qualitative and follows the scope of (Ramos et al., 2020), namely autonomous ships operation and collision scenarios. Nonetheless, it can be expanded to other autonomous systems' operation.

This paper is organized as follows. Section 2 provides and overview of the H-SIA method. Section 3 introduces the extension to H-SIA with Fault Trees. H-SIA is applied to a case study in Section 4. Section 5 presents concluding remarks about the method and its application.

2. H-SIA METHOD ELEMENTS

The HSIA method is comprised initially of two main elements: the ESD and the CoTA (Figure 1). While the ESD models the events that can take place following an initiating event, the CoTA models which tasks the agents must perform for these events to succeed. In other words, the ESD models *what* can happen, while the CoTA models *how* the agents can reach successful outcomes (Ramos et al., 2020). The extension of H-SIA addresses how the agents can *fail*. The ESD, the CoTA, and an overview of the FTs are briefly described in this section. For a detailed description of the ESD and the CoTA the reader is referred to (Ramos et al., 2020). The extension of H-SIA, comprised of the generic FTs and basic failure events, is detailed in Section 3.

2.1 SCOPE OF DEVELOPMENT

H-SIA was developed for autonomous ship operation and collision scenarios. Collision may be immanent if a vessel is heading towards the own vessel and does not change course, or if a vessel changes course that is large or in tow (International Maritime Organisation, 1972). Following (Li et al., 2012; Otto et al., 2002), it is considered that the autonomous ship is on collision course when it is in a situation in which it will collide with the collision candidate (CC) if the course / speed of the vessel(s) is not changed.

The autonomous ship considered for the development of H-SIA is an unmanned ship. The autonomous ship may be supervised or remote-controlled by operators in a Shore Control Center (SCC). The SCC crew is composed of teams. An operator may be responsible for monitoring more than one

autonomous ship at a time, and an experienced shift supervisor may be available. The collision candidate may be an autonomous ship, a conventional ship, a fixed or a floating object.

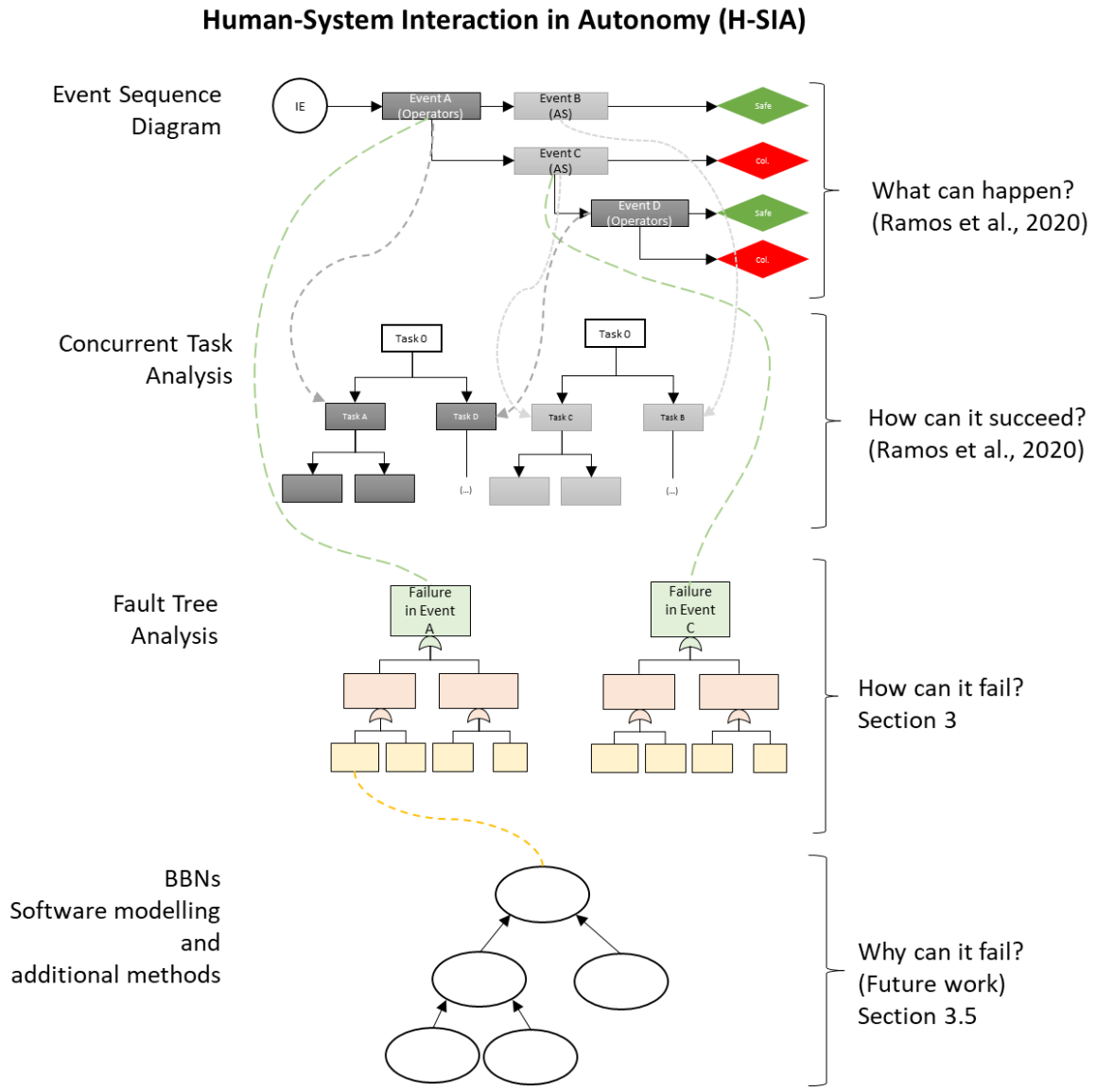

FIGURE 1: OVERVIEW OF THE EXTENDED H-SIA METHOD.

2.2. SCENARIO MODELING: EVENT SEQUENCE DIAGRAM

An ESD represents the possible sequence of events following an initiating event, e.g., a disturbance of the normal state of the system, leading to the possible final consequences (end states) (Rausand, 2011; Swaminathan and Smidts, 1999a, 1999b). As H-SIA initially focuses on accident scenarios of autonomous ship, the initiating event is the ship being on collision course. H-SIA provides a flowchart to develop ESDs for autonomous ship of different designs and LoAs. The ESD flowchart captures autonomous ships with different LoAs ranging from low LoA, remotely controlled by operators working in a SCC, as well as a fully autonomous ship, not supervised by any human. The flowchart is also applicable in case the LoA is changing dynamically. Furthermore, it includes possible recovering

events. For instance, in case the autonomous ship is responsible for detecting a collision candidate and fails to do so, the flowchart guides the analyst to include the event of an SCC operator (if applicable) performing the detection. The flowchart ensures traceability and reproducibility. The analyst's assumptions and considerations about the design and operation of the system are clearly documented.

2.3 MODELING SUCCESS PATHS: CONCURRENT TASK ANALYSIS

The CoTA was introduced with H-SIA to analyze complex systems. It models the interactions between different agents (e.g., humans and autonomous ship) on a task level, and it builds upon Task Analysis theory (Annett and Stanton, 2000; Diaper and Stanton, 1992; Shepherd, 2001). The CoTA comprises a Task Analysis of each agent of the system, executed concurrently. For the CoTA development, the events of the ESD are translated into high-level goals (e.g., detect collision candidate). Those are re-described (decomposed) into basic tasks.

An important feature of the CoTA is the rules for task re-description. They use the IDA (Information, Decision and Action) cognitive model (Shen et al., 1997) and extend the model to the whole system (Figure 2). An agent receives information from the external world, e.g., an alarm, in case of an operator, or signals and data, for an autonomous ship system. This information is processed through the (I) block. During the decision phase (D) the agent responds to the information, through situation assessment, diagnosis and response planning. The decisions made in the (D) phase are put into action through the (A) phase (Chang and Mosleh, 2007a). Typically, the operator performs an action on an external world, such as a screen or hardware, whereas a component of the system may perform an action on another component or on the external world. The IDA process is influenced by a "mental" state, for the operators, or the system models, for the autonomous ship. For the operator, the mental state combined with memory represents the operator's cognitive and psychological states. For the autonomous ship, the autonomous ship model includes the programmed behavior, the process models in the autonomous ship, and the world model of the autonomous ship.

The advantage of using the IDA phases is that, for the operator, failures can be further traced to the cognitive process leading to an error. For the system, failures can be traced to the responsible component in which the they occurred (e.g., an error in information retrieval can be traced to the responsible sensor). Moreover, performing task analysis in parallel for several agents concurrently, as in the CoTA, ensures that low-level tasks from all agents are in the same level of re-description.

Following the re-description of the tasks using the IDA model, the CoTA rules enable the analyst to identify and re-describe "interface" tasks. These interface tasks are tasks that need an input from or give output to another agent. Consequently, the CoTA identifies not only the tasks that must be performed by an agent, but the interactions between agents and their task dependencies.

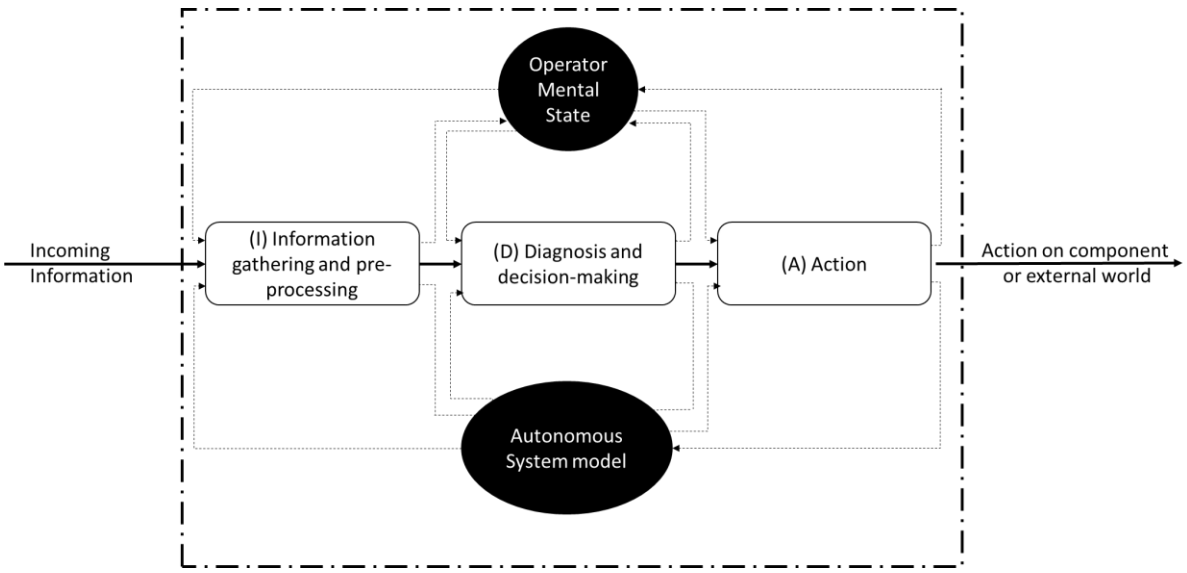

FIGURE 2: INFORMATION, DECISION AND ACTION MODEL USED IN H-SIA (ADAPTED FROM (Ramos et al., 2020)**).**

2.4 MODELING FAILURE PATHS: GENERIC FAULT TREES

The extension of H-SIA, described in this article, addresses the need for a more thorough analysis of failure causes to complement the success-oriented view of the CoTA. A more detailed failure analysis allows for identifying risk reduction measures, "weak links" and insufficient design solutions in the system. FTs need to represent the human operator or supervisor in the SSC, as well as the autonomous ship (Utne et al., 2017).

The FTs of H-SIA provide a skeleton for failure analysis of a task and a basis for comparable results. The generic fault trees for the autonomous ship and human operator related events build on the IDA structure. They guide the analyst into identifying basic failure events that occur in one of the IDA phases. Since the presented FTs' elements are generic, the analyst should identify and include only the relevant branches for the system and scenario being analyzed. According to the IDA model, human operators or the autonomous ship can fail during:

i) Information gathering and pre-processing (I phase), for example: receiving information from sensors (autonomous ship), responding to an alarm (SCC);
ii) Situation assessment and decision making (D phase), for example: deciding a new course for collision avoidance through the correct algorithms (autonomous ship) or correct mental model (SCC);
iii) Action taking (A phase), for example: sending command to correct thrusters (autonomous ship); sending command through the HMI (SCC).

The FTs also address the propagation of failures arising through the interaction between the system's agents, identified in the CoTA as interface tasks. In addition, the FTs include the failures in the parallel tasks identified in the CoTA. Parallel tasks are supporting tasks, i.e., they are necessary for the successful execution of the other tasks and the interaction between the agents. Moreover, parallel tasks are related to the normal operation of a system and do not follow a specific order in a plan, i.e., they are executed at the same time as other tasks (Ramos et al., 2020). This is particularly relevant for the autonomous ship's tasks *Information collection* and *Establishing communication*. Specialized FTs were deemed necessary to provide analysts with guidance on these two complex events, such that

they are comparable and reproducible for different design solutions. The process for developing the generic FTs are provided in the next section.

3. DEVELOPMENT OF GENERIC FAULT TREES FOR AUTONOMOUS SYSTEMS

The development of FTs from the CoTA and the derived Basic Failure Events are the main addition to H-SIA and are described in detail in this Section. We propose generic FTs that lead to the identification of generic basic failure events[1]. We also present the possibilities of further analysis with Bayesian Belief Networks (BBNs), software risk models, or additional methods in section 3.4. The extension of the H-SIA method allows for improved analysis of the autonomous system in terms of failure behavior, which has not been addressed in detail in Ramos et al. (2020).

3.1 H-SIA FAULT TREES OVERVIEW

FT analysis is a deductive methodology, to model Boolean combinations of basic failure events leading to a critical event in a system, in a top down logic. Basic failure events are the lowest events considered in the FTs and represent the resolution of analysis.

The FTs guide the analyst through generic basic failure events. Note that the basic failure events may describe failure modes, failure causes, or failure mechanisms. There is no consensus in the use of these terms among different disciplines, e.g. in human reliability and software reliability. This paper adopts the term “basic failure events”, which derives from FT analysis and captures best the intended meaning. The basic failure events cover hardware, software and human failure. Hardware failure refers to the failure of physical components of the system, these may be mechanical, electrical or electronical components, for example, pumps, engines, sensors, computing hardware, or transmitters. It is noteworthy that a failure of the computing hardware may lead to a software failure. Therefore computing hardware failures may need to be considered for software failure analysis, see Section 3.5 (Ozarin, 2013; Thieme et al., 2020a). Software failure is the loss of a functionality occurring through the output of software, this output may be provided or not provided in time (Ozarin, 2009). A software failure may arise through failures in the specification, design, implementation, data entry faults, or computing hardware errors (Guarro et al., 2013; Ristord and Esmenjaud, 2002; Thieme et al., 2020a). Human failure (or human error) may receive different definitions according to methods and authors. Examples of human error classifications include intentional and unintentional errors, errors of omission and commission (Swain and Guttmann, 1983), skill-, rule-, and knowledge-based errors, and slip, lapse, mistake, and violation (Reason, 1990) [5], among others. This paper adopts IDAC’s classification of human error (Chang and Mosleh, 2007a): an observable human action can be classified as an error with respect to the external reference points: the system, procedures, and the crew. A mismatch between the states and mutual requirements of any two of these reference points can be classified as an error.

[1] In H-SIA, the term “failure event” refers to the failure in the events identified in the ESD, whereas “basic failure event” prescribes an explanation for the cause of failure. For instance, “failure in detecting collision candidate” is a failure event identified at the ESD, and “failure from the sensors in collecting raw data” is a basic failure event.

The structure of the generic fault tree for the human operator uses as a foundation a model-based Human Reliability Analysis (HRA) method (Hendrickson et al., 2010), further advanced in the Phoenix HRA method (Ekanem et al., 2015). The development of those leveraged from the Task Analysis of the system developed in (Ramos et al., 2020). The generic fault trees for the autonomous ship follow the same structure, considering the possible failures concerning information collection, decision-making, and action taking. The IDA model provides a structure that is independent of the system architecture and therefore can accommodate a variety of designs. Some of the basic failure events are named similarly for the autonomous ship and the human operators, as it describes a similar mechanism of failure with basis in the IDA model.

The events in the FTs are connected through the standard logic gates (*and*- and *or*-gates). The FTs also include "flags", which indicate that a branch of the FT may be neglected according to the scenario analyzed. For instance, if the event analyzed does not include an operator in the SCC sending a command to the autonomous ship, the branch related to a failure in the autonomous ship in receiving and implementing this command is not relevant for the analysis. In this case, this would be indicated by "flag of no command / request". In simple terms, a flag assigns the probability 1 to an event gate.

3.2 DEVELOPMENT OF H-SIA FAULT TREES

Two features of the CoTA drive the FT development: the interface tasks and the parallel tasks. The interface tasks are those that connect the different agents through the task level. An example of an interface task is to execute an operator command by the ship. The system, firstly, must correctly receive the command. "Implement a command" is thus an interface task of the system, and "send a command" is an interface task of the other agent, e.g. operator. They are connected and are dependent in a task level. The cause of a failure in one agent's interface task may be due to, among others, a failure in the interface task of the other agent that should provide an input to the first one. The interface tasks are modeled through their own FTs, which are connected with an or-gate to the other agents' FTs if applicable.

Parallel tasks are tasks that support other tasks. An example of a parallel task is the ship task of "Collecting data". The sensors and relevant hardware and software must collect data of the ship and its surroundings during the whole operation. A failure in the parallel tasks is modeled through its own FT, also following the IDA model. Because the parallel task supports the execution of the other tasks, a failure in the parallel task may lead to a failure in the other tasks. The FT of the parallel task is thus connected through an or-gate to other FTs, when applicable.

The separately modelled FTs for parallel and interface tasks ensure the inclusion of the dependency of failure on a task level. Figure 3 illustrates the relationship between the FTs, the CoTA and the ESDs. The numbered arrows represent the steps for the development of the FTs in H-SIA:

(1) Each event of the ESD is re-described in the responsible agent's CoTA;
(2) The failure in performing the event is the top event of a FT. The failure event can be due to a failure in collecting necessary data, and/or a failure in making the correct decision, and/or a failure in action taking. Note that an error in action that derives from an error in decision-making is defined as a correct action given an incorrect decision.
(3) The parallel tasks are modeled through their own FTs. They should also follow the IDA phases as paths leading to failure event. They are connected to the main FTs through gates, when applicable;

(4) The interface tasks that sends in input to another agent are modeled through their own FTs. They are connected to the main FTs of the agents through gates. In Figure 3, the AS sub-task 1.2 should receive an input from the human. This human task is modeled through its own FT and connected to the AS FT.
(5) The FTs lead to basic failure events. These can be defined generically in cases the specific features of the system are not defined. Figure 6 illustrates this for the case for sensor failure.

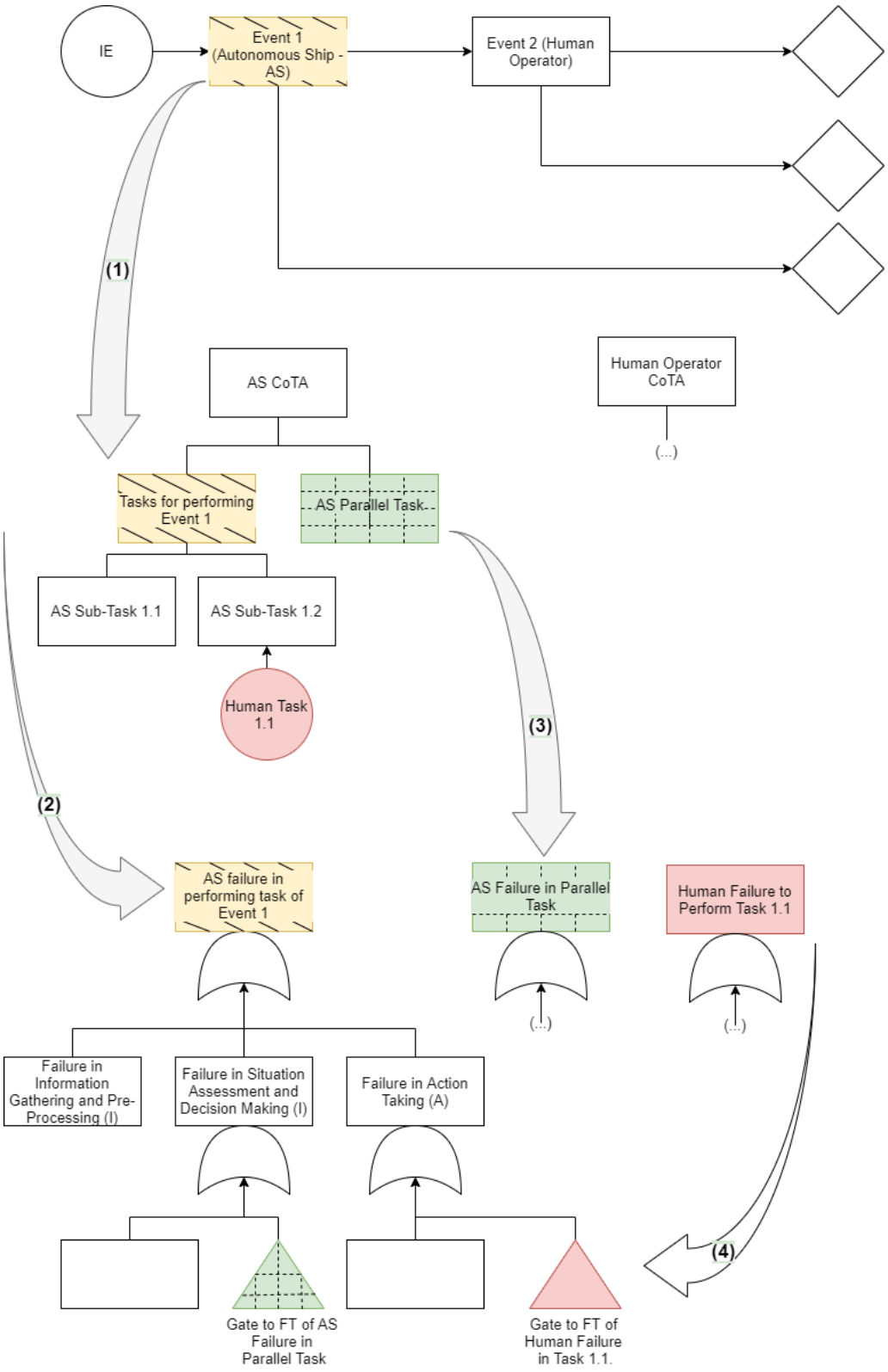

FIGURE 3: DEVELOPMENT OF THE FTS THROUGH THE ESD AND THE COTA

The following sections provide FTs for a generic autonomous ship and for humans working in a generic SCC during a collision avoidance event. The FTs were developed using the abovementioned rules and

can serve as a basis for FT development for similar systems. The fault trees are based on the following assumptions:

i) The autonomous ship is unmanned;
ii) According to the LoA, the ship may be supervised and/or remotely controlled by operators working in a SCC.

3.3 GENERIC FAULT TREES FOR THE AUTONOMOUS SHIP

The generic structure of the autonomous ship FTs combines software-related, computing-hardware-related and physical system-related events as failure causes. The FTs require knowledge of the autonomous ship in order to choose the applicable branches and basic failure events. Furthermore, some of the basic events in the FTs are not fully developed, instead, they intend to remind the analyst that design-specific basic events should be included in this branch, as described in the previous section. This allows the FTs to accommodate different systems designs. For instance, one of these events is "sensor failure". It can be further developed to indicate which type of sensor and which type of failure may happen (e.g., dirty camera lenses or inaccurate radar). These events are indicated in the FTs with dashed borders.

The top event of the autonomous ship FT is a failure event that is the responsibility of the autonomous ship in the ESD. It is generically named here "autonomous failure event". An autonomous failure event corresponds to one of the events in the ESD, for example, Failure in Detection by autonomous ship. This example is further explained in Section 4. The failure may take place due to Failure in collecting necessary data (I Phase) or failure in making the correct decision, or failure in taking/ executing the correct action (see Figure 4).The separation into these three types fits the view taken in standards for safety critical systems, such as, IEC61508 (IEC, 2010). In IEC61508 the system may be separated into three types of components: Sensor system, logic solver, actuators/final elements.

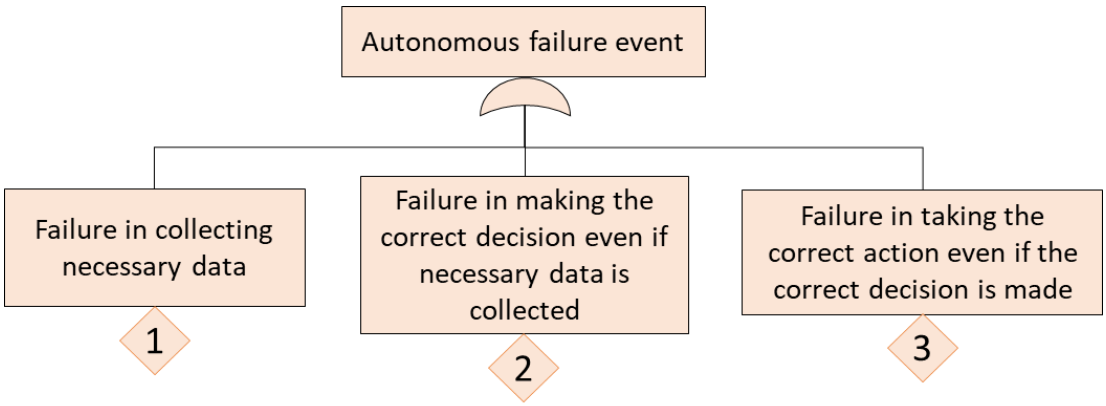

FIGURE 4 TOP EVENT FOR THE GENERIC AUTONOMOUS FAILURE EVENT FAULT TREE.

The autonomous ship performs two parallel tasks: *data collection* and *communication. Data collection* is essential for the ship operation. It refers to the collection of all necessary internal and external data, including environmental conditions, objects in proximity, machinery performance data and navigational data. Data collection can fail in two ways: the ship may not collect the necessary data, e.g., due to sensor failures; or the ship may collect incorrect or incomplete data. *Communication* refers to the communications links between the ship and the SCC and other traffic participants. This includes

sending data to the SCC and receiving data / commands from operators. *Communication* has a higher importance if the ship is monitored or controlled by operators onshore.

As mentioned previously, failure in parallel tasks are modeled through exclusive FTs, which also follow the IDA model. They are connected to the main FTs of the ship and the human operators through Boolean Or-gates. The autonomous ship' FTs of the IDA phases are described below, followed by the failures in the parallel tasks, abbreviated as SDC-N (Ship Data Collection -No Data Collected), SDC-I (Ship Data Collection – Incorrect Data Collected), and SCF (Ship Communication Failure). The basic failure events are described in the Appendix.

3.3.1 AUTONOMOUS SHIP FAILURE IN COLLECTING NECESSARY DATA - I PHASE

The autonomous ship is equipped with sensors and related equipment for collecting data about the ship itself and the surroundings. In case there is a SCC, the operators may send a request to the autonomous ship for more information, or a command. The autonomous ship has thus two sources of data. The system may fail in data collection due to (Figure 5):

i) No data is collected. Ship Data Collection – No data (SDC-N) is modeled through its own FT and connected through an Or-gate;

ii) Incorrect data is collected. Ship Data Collection – Incorrect data (SDC-I) is modeled through its own FT and connected here through an Or-gate;

iii) Incorrect or no command is sent by the humans due to:

   a. Failure in communication with the SCC. Ship Communication Failure (SCF) is modeled through its own FT and connected through an Or-gate;

   b. Human failure (HFA – Human Failure in Action): the operator fails to send the correct command, or to send it in time. This failure is modeled through its own FT.

   Note that if the event does not involve a command / request from the SCC, the flag should be used.

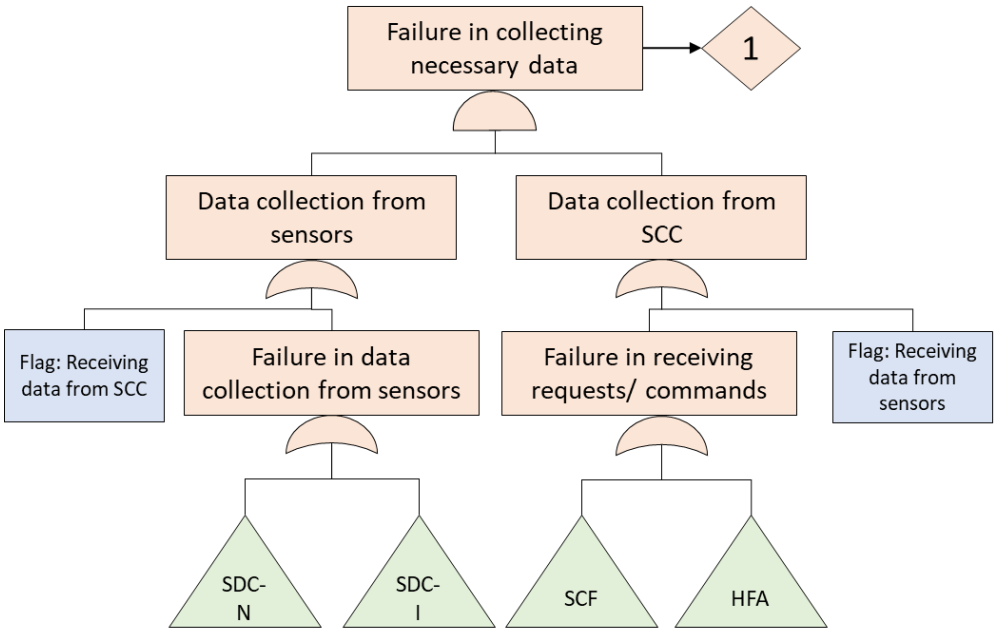

FIGURE 5 GENERIC FAULT TREE FOR THE AUTONOMOUS SHIP FAILURE IN COLLECTING NECESSARY DATA

3.3.2 AUTONOMOUS SHIP FAILURE IN DATA COLLECTION - NO DATA COLLECTED (SDC-N)

The failure in data collection resulting in no data being collected may be due to (Figure 6 and **Error! Reference source not found.**Table 2 in the Appendix):

i) Failure in collecting raw data – This may be a failure in the sensors and databases (solution specific basic failure events should be included);

ii) Failure in the planning to collect information, due to:
   a. Inadequacy of the data sampling frequency;
   b. Data may not be identified as needed to be polled;
   c. Data may be discounted, e.g., through a weighing process.

iii) Failure in execution and to collect data, due to:
   a. Failure in the support system;
   b. Data failures and data limitations of the system.

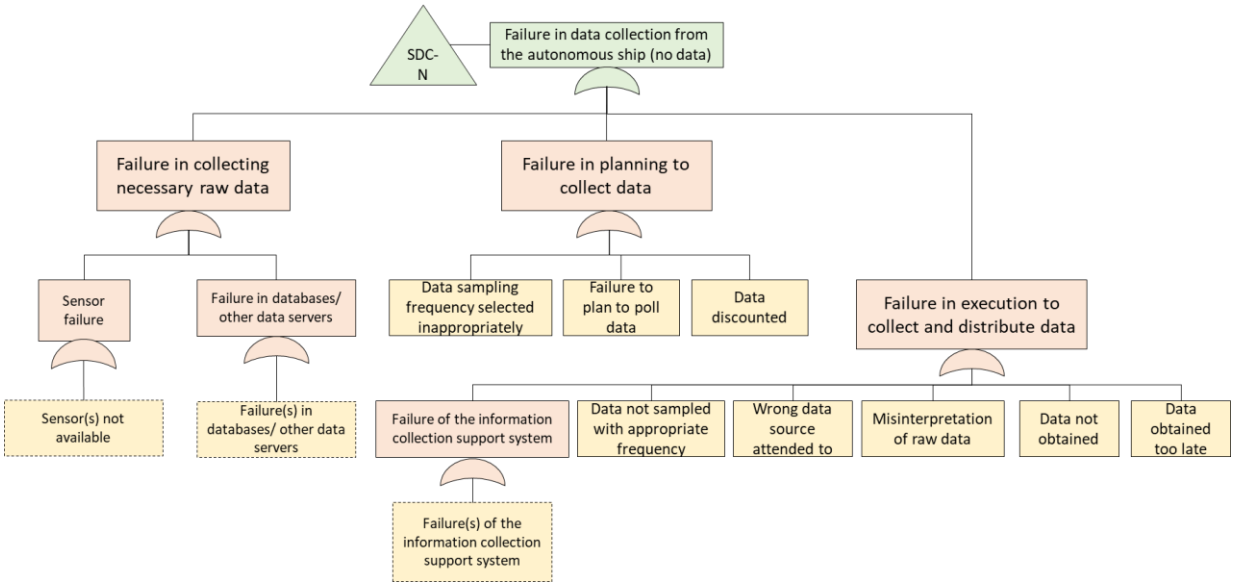

FIGURE 6 GENERIC FAULT TREE FOR THE AUTONOMOUS SHIP FAILURE IN DATA COLLECTION - NO DATA (SDC-N)

3.3.3 AUTONOMOUS SHIP FAILURE IN DATA COLLECTION - INCORRECT DATA COLLECTED (SDC-I)

Similar to the failure of no data collection, the system may collect incorrect data and use it in the further process of decision-making. The tree and its reasoning behind are very similar to the considerations presented for the tree of not delivering any data. The main difference is the "AND" – gate on the top, which combines that the autonomous ship needs to collect wrong data and not realize that it collected the wrong data (Figure 7).

Incorrect data may be collected through failures in sensors, incorrect database entries, the decision to collect incorrect data, or a failure on the action of collecting data (Figure 7). Among others, the data sources also may be incorrectly polled, e.g., for certain data, or the system may generate incorrect or noisy measurements. If the system does not recognize this data as incorrect it will use the data for further decision-making and actions and hence fail. The failure of the autonomous ship to collect data can be due to no data being collected, or incorrect data being collected. The basic failure events for those are very similar and are therefore described together if in Table 2 in the Appendix. Specific references for no data collected or incorrect data will be marked with *SDC-N* or *SDC-I*, respectively.

3.3.4 AUTONOMOUS SHIP FAILURE IN COMMUNICATION (SCF)

The failure in communication establishment between the SCC and the autonomous ship may be due to the following events (Figure 8). The associated failures can be found in Table 3 in the Appendix.

i) Failure in receiving request - There may be an error in information retrieval, in this case the request to establish a communication or data link cannot be received. Note that this branch should be ignored if the scenario analyzed does not include a request for communication.

ii) Failures in decision making with respect to communication and data transfer. These may arise from choosing the wrong communication channels or partners, or not being able to process, retrieve the necessary information, or prioritization of the execution of other actions.

A failure on the action side of establishing communication and data links can be caused by a failure in the hardware or software, or incorrect operation of the communication equipment, timing related failures or incorrect establishing of communication between different partners.

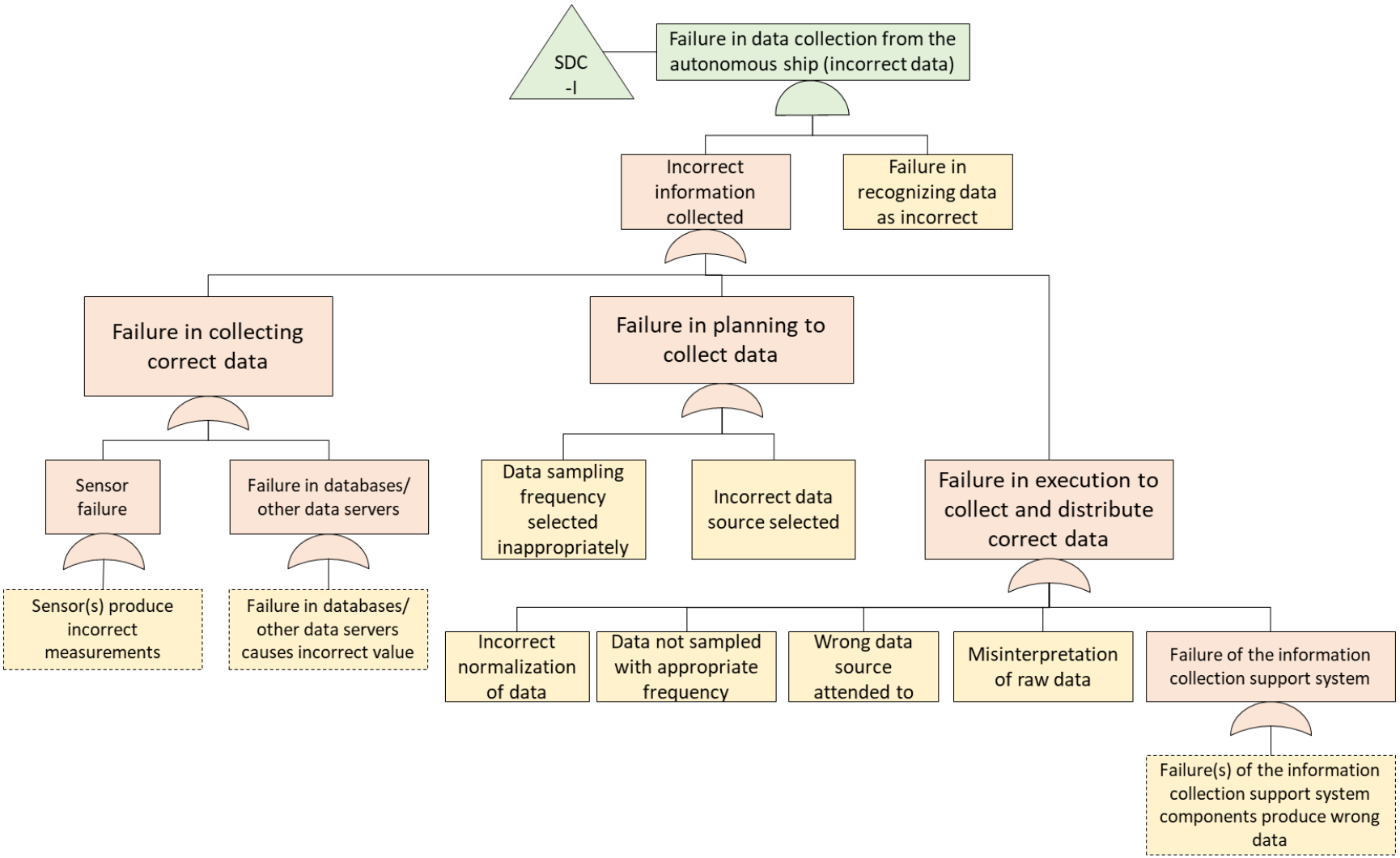

FIGURE 7 GENERIC FAULT TREE FOR THE AUTONOMOUS SHIP FAILURE IN DATA COLLECTION - INCORRECT DATA (SDC-I)

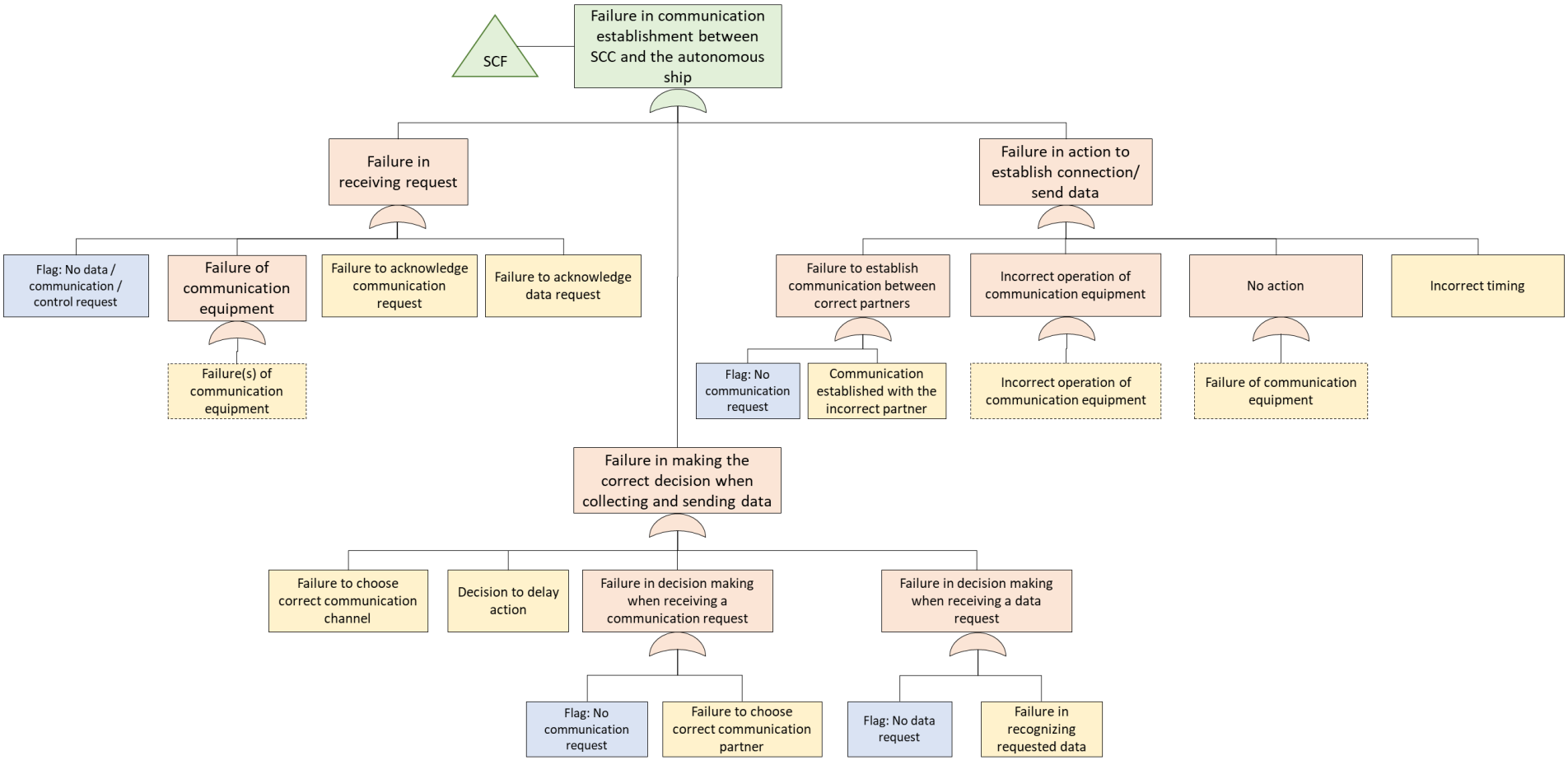

FIGURE 8 GENERIC FAULT TREE FOR THE AUTONOMOUS SHIP FAILURE IN COMMUNICATION ESTABLISHMENT BETWEEN SCC AND THE AUTONOMOUS SHIP (SCF)

3.3.5 AUTONOMOUS SHIP FAILURE IN SITUATION ASSESSMENT AND DECISION MAKING - D PHASE

The autonomous ship' failures in making the correct decision are general failures to arrive at a satisfactory decision on an action in a given situation. That implies that there is a decision that can avoid or mitigate a situation, e.g., avoided consequences, least damage possible, or least severe consequences. For the autonomous ship, this failure may be due to (Figure 9 and Table 4 in the Appendix):

i) The state of the system and surroundings is misdiagnosed by the autonomous ship;
ii) The autonomous ship may fail to adapt the procedure to the current situation;
iii) The autonomous ship may decide on an infeasible/ inadequate solution (strategy) or delay further action, e.g., keep course;
iv) The system may transfer to an inadequate procedure, e.g., apply the wrong COLREG rule.

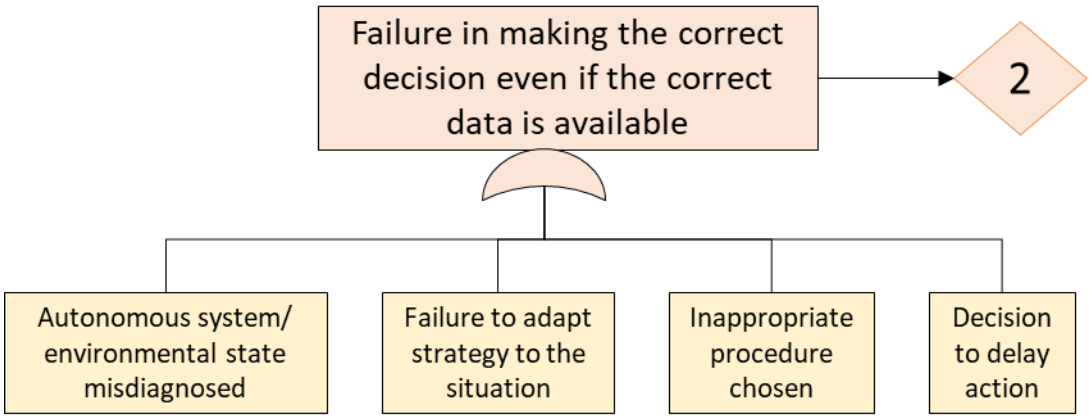

FIGURE 9: GENERIC FAULT TREE FOR THE AUTONOMOUS SHIP FAILURE IN MAKING THE CORRECT DECISION

3.3.6 AUTONOMOUS SHIP FAILURE IN ACTION - A PHASE

A failure for the ship to act (Figure 10) may have roots in failures of the computing hardware, software or general the ship hardware. Basic failure events may be engagement of the wrong actuators, e.g., wrong thrusters, or the timing of engagement may be inadequate (a maneuver stopped too early, or an action initiated too late). A component may be operated inadequately, or no action may be executed due to failure of components. Table 5 in the Appendix present the Basic Failure Events description.

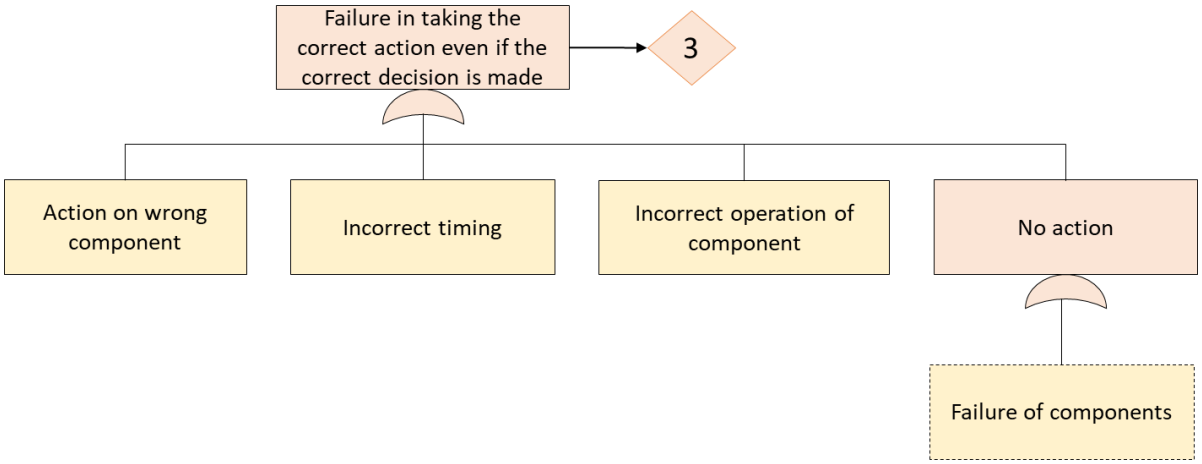

FIGURE 10 GENERIC FAULT TREE FOR THE AUTONOMOUS SHIP FAILURE IN TAKING CORRECT ACTION

3.4 GENERIC FAULT TREES FOR HUMAN OPERATORS

The generic FTs of the human operator builds on the FTs of the Phoenix HRA methodology (Ekanem et al., 2015), based on the HRA framework (Hendrickson et al., 2010; Mosleh et al., 2010). The FTs were developed from cognitive science literature and were originally developed for operators working in a control room of a nuclear power plant (NPP). This type of operation shares similarities with operation of unmanned autonomous systems, whose operators will be monitoring / controlling the system remotely, probably working in a control center (Man et al., 2015; Peter Barthelsson and Sagefjord, 2017; Porathe, 2014; Ramos et al., 2019). The operators of both NPPs and autonomous systems should be highly trained, should have procedures / guidelines for their operation, and may work together with other crew members and/or supervisors. The structure and basic events of the FTs of Phoenix are thus highly applicable for autonomous systems. Nonetheless, the FTs in Phoenix must be adjusted to reflect the particularities of operators interacting with autonomous systems.

One of the particularities of unmanned autonomous ships operation is that the ship would be operated remotely, from large distances (Kari et al., 2018)[2]. Contrarily to other operations in control rooms, such as NPPs or the process industry operations, open sea operations where the autonomous ship is unmanned and difficult to access, there are no "field operators" who could personally interfere in its operation. For the SCC to receive information about the ship and its operational environment, or to send commands or requests, the communication link must be established successfully. Operators will not have the opportunity to collect data, make decisions and take actions if there is no communication. Therefore, in addition to failures in the I, D and A phases, the Human Failure Event may also occur due to a failure in communication establishment between SCC and the autonomous ship (gate to the FT of Ship Communication Failure – SCF) (Figure 11):

The FTs corresponding to the IDA phases are described below. Note that the failure in taking the correct action is modelled through its own FT because this is an interface task. The operators perform an action in the HMI, sending a command to the autonomous ship. This may be a change in the heading using a joystick, or a command for receiving more data on a specific object, among others. A failure in

[2] Some autonomous ship projects include the possibility of having manned autonomous ships, in which the operators would be onboard. H-SIA was not developed for this case.

operators' actions can lead to the human failure event, but also affects the autonomous ship operation. When the operators fail to send a command or send an incorrect command, it affects the autonomous ship's execution of this command. The FT of the HFA is connected, thus, to the fault trees of human failure event (Figure 11) and to the FT of autonomous failure event, through a gate in the failure in collecting necessary data (Figure 5).

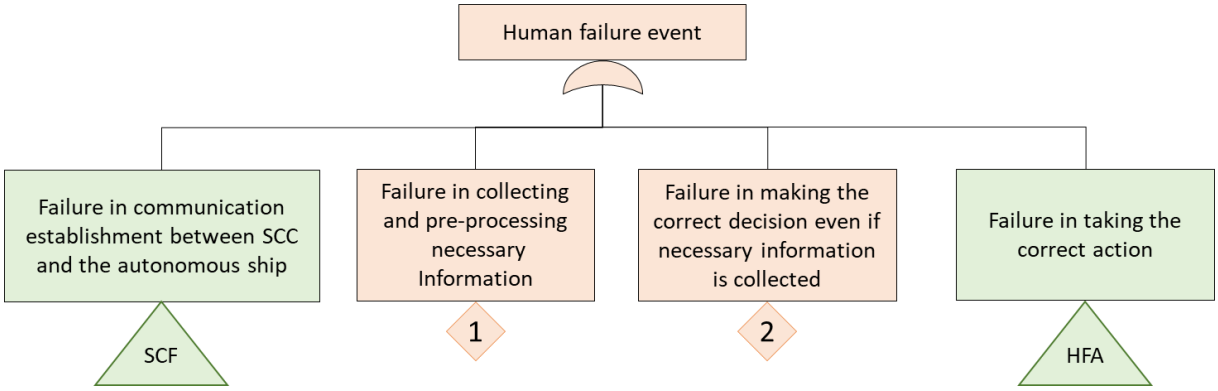

FIGURE 11: GENERIC FAULT TREE FOR HUMAN FAILURE EVENT

3.4.1. OPERATOR'S FAILURE IN INFORMATION COLLECTION AND PRE-PROCESSING - I PHASE

The operator can fail in collecting and pre-processing the necessary information due to (Figure 12):

i) No information is received from the autonomous ship. Note that this does not concern the failure in communication establishment. Instead, it covers a failure in the HMI, which is the main source of information for the operator, or the failure in the data collection from the autonomous ship;
ii) Failure in collecting correct and complete data due to a failure in the information source and the failure from the operator to recognize the information as incorrect;
iii) Failure in decision to collect information (Figure 13);
iv) Failure in execution to collect information (Figure 14).

The failure in information source considers all sources the operator could use. All these sources must fail for the failure in information source to take place: the operator may collect incorrect data from one source but correct one from another. There is the possibility that one or more of these sources are not available or not applicable – this is indicated with the flags.

.

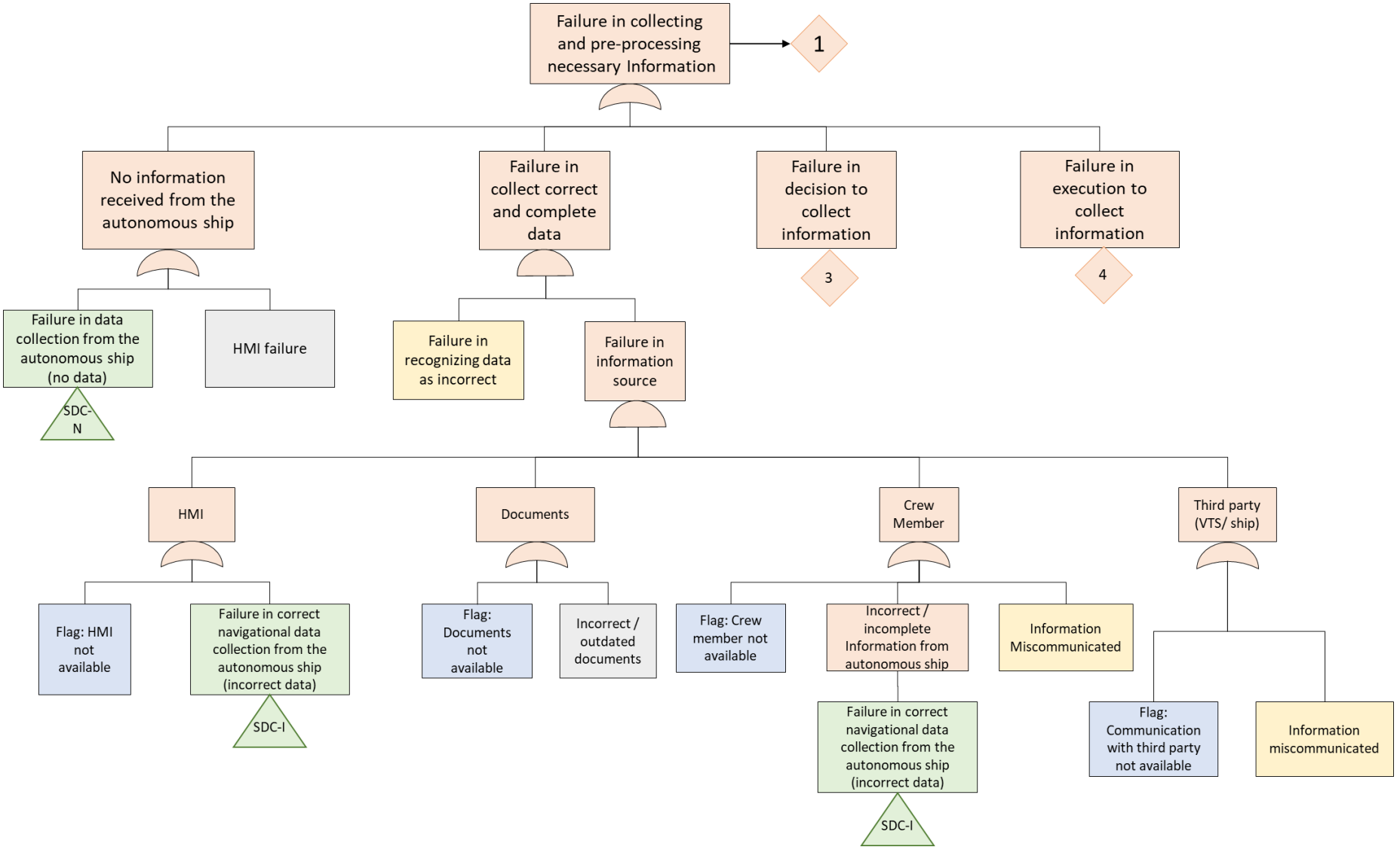

Figure 12: **GENERIC FAUL TREE FOR HUMAN FAILURE IN COLLECTING AND PRE-PROCESSING NECESSARY INFORMATION**

The failure in decision to collect information may happen when the operator is following a procedure, guideline or any other written rules as strategy, and/ or when the operator is following her/ his own knowledge. Furthermore, the SSC operator may decide to collect the necessary information but fail in executing it (Figure 14).

Table 6 in the Appendix presents the Basic Failure Events description for the human failure in information collection and pre-processing.

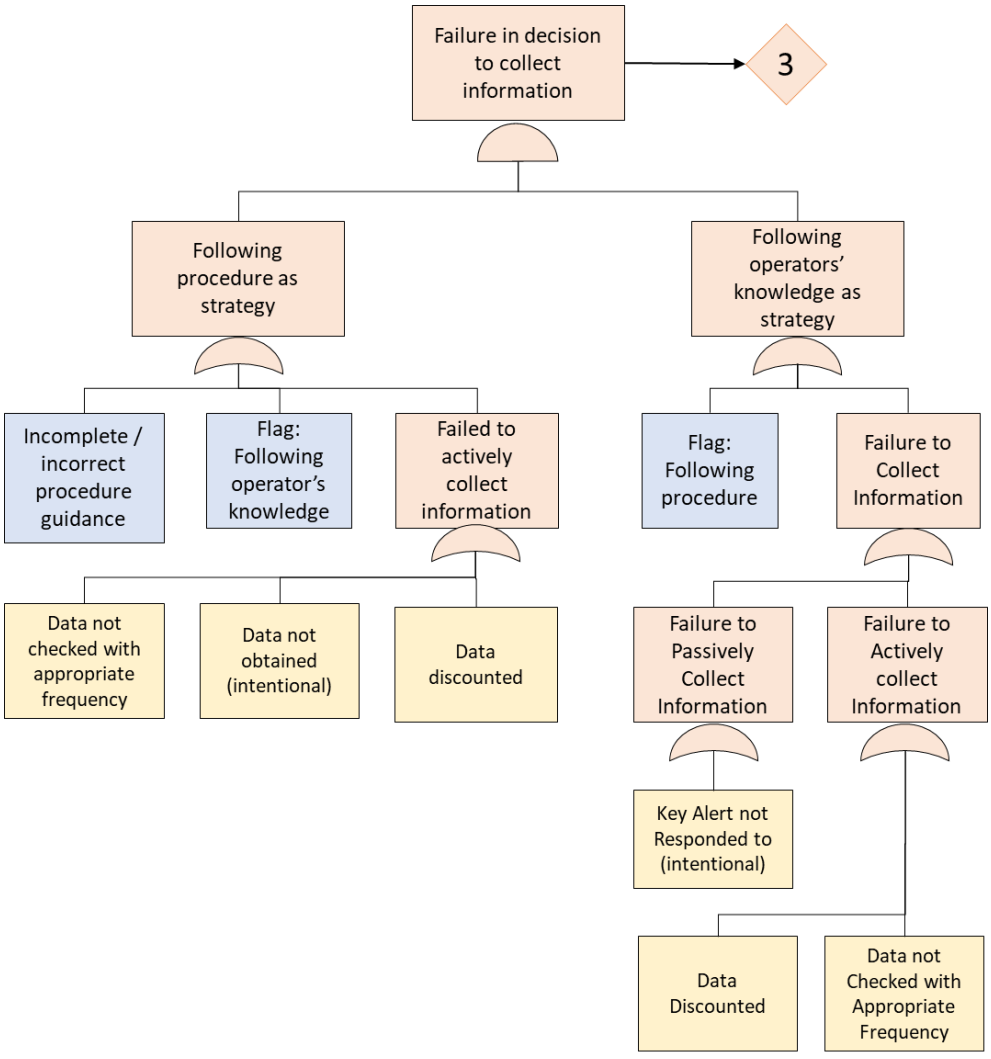

FIGURE 13: GENERIC FAULT TREE FOR HUMAN FAILURE IN FECISING TO COLLECT INFORMATION

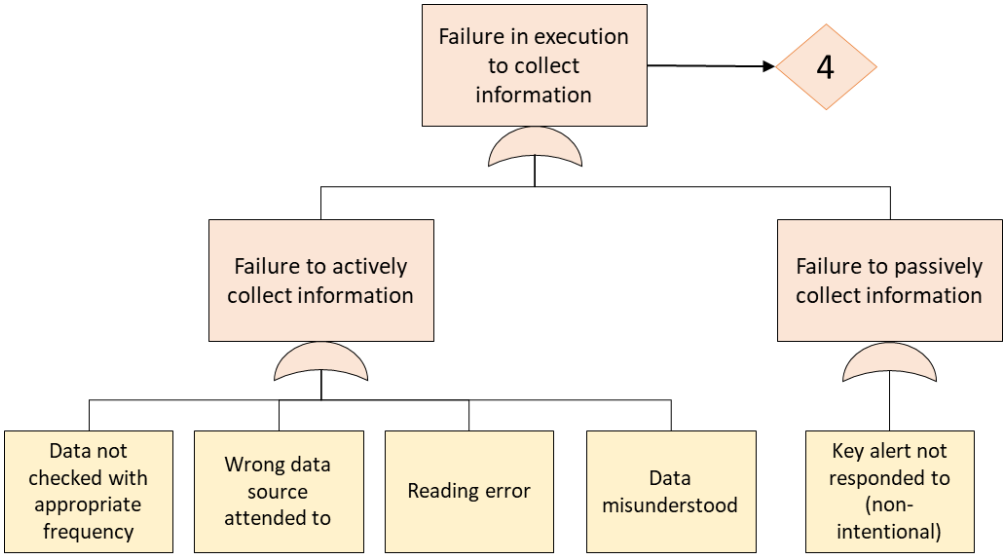

FIGURE 14: GENERIC FAULT TREE FOR HUMAN FAILURE IN EXECUTING TO COLLECT INFORMATION

3.4.2 OPERATOR'S FAILURE IN SITUATION ASSESSMENT AND DECISION MAKING - D PHASE

Failures in situation assessment and decision making may occur when the operator is following a procedure / manual / guideline, and / or when the operator is relying on knowledge. In the FTs the term "procedure" may refer to any written guideline, including those that describe the actions the ship should take, such as COLREGs and local rules, and those that guide the operator on how to interact with the HMI and the autonomous ship. The branches of the FT concerning the failure in the D phase are presented in Figure 15 and Table 7 in the Appendix.

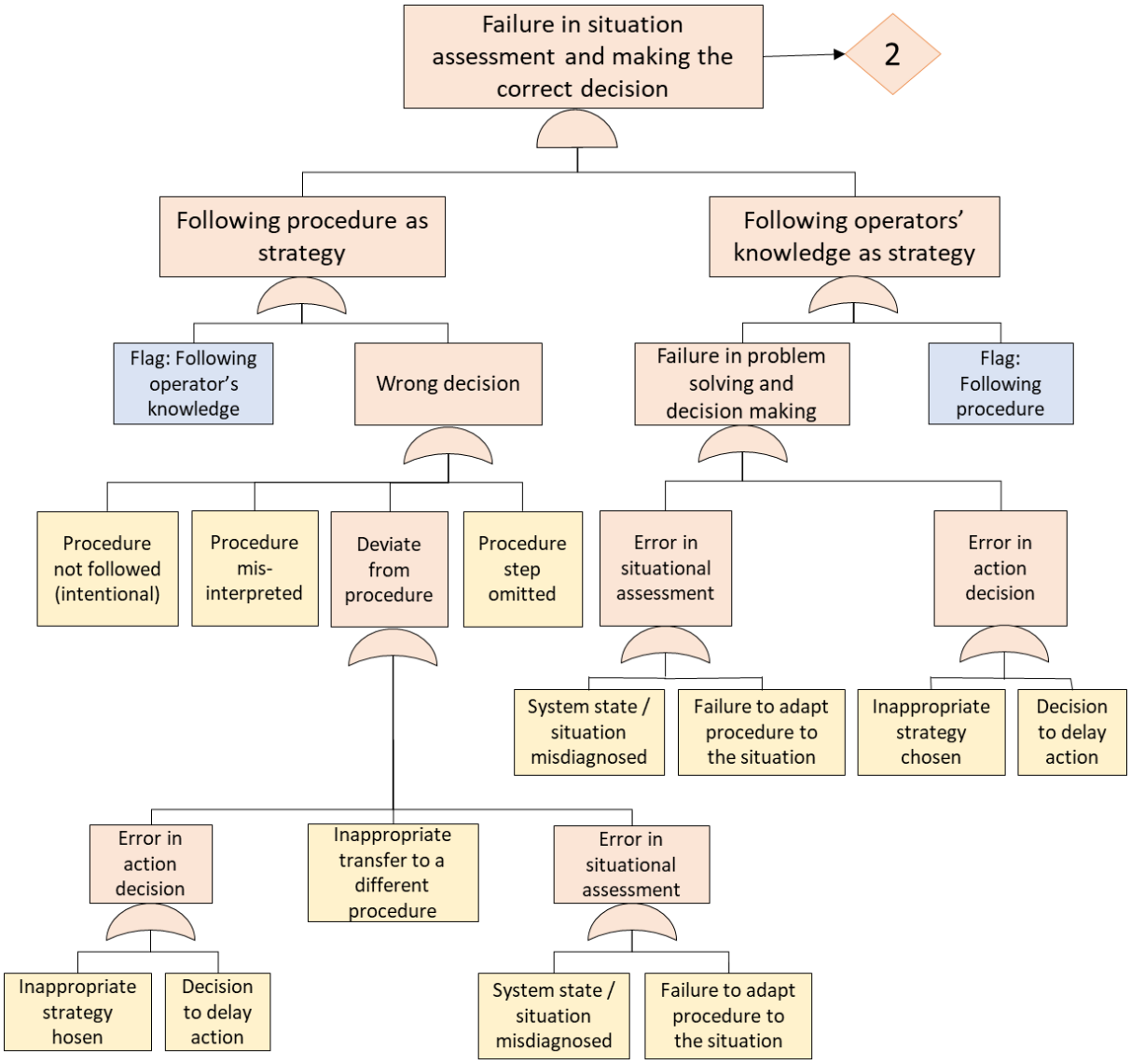

FIGURE 15: GENERIC FAULT TREE FOR HUMAN FAILURE IN SITUATION ASSESSMENT AND DECISION-MAKING

3.4.3 OPERATOR'S FAILURE IN ACTION - A PHASE (HFA GATE)

Having made a correct decision, the operator may fail in executing it. Note that if the operator makes an incorrect decision, e.g., to press a certain button s/he should not, and executes it, the failure is in the Decision phase, and not in the Action phase. Failure in the A-Phase are thus a failure in correctly performing an action that follows a correct decision. The operator may fail in this phase because (Figure 16 and Table 8 in the Appendix):

1. The operator may act on the wrong component / object;
2. The operator may act with incorrect timing;
3. The operator may act on the correct object, but in an incorrect way.

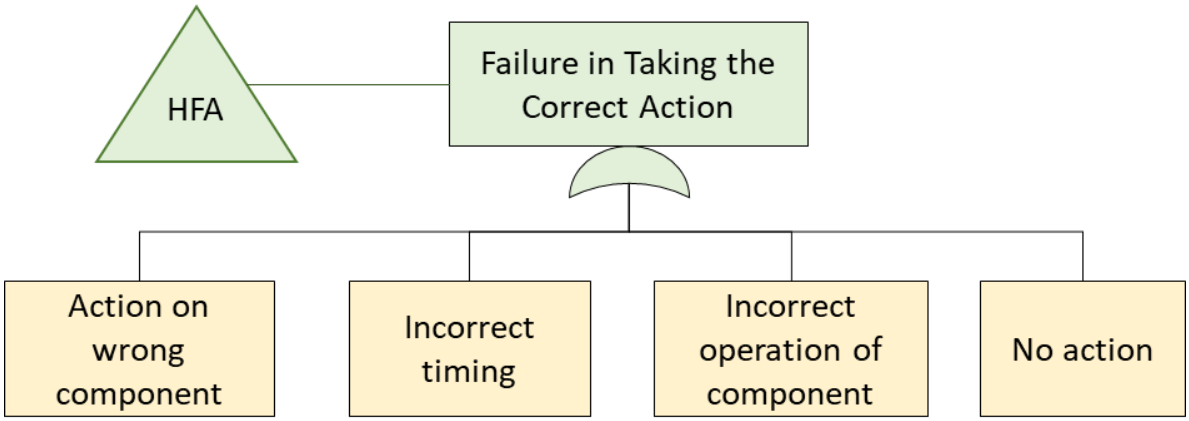

FIGURE 16: GENERIC FAULT TREE FOR HUMAN FAILURE IN TAKING THE CORRECT ACTION (HFA)

3.5 POSSIBILITIES OF FURTHER ANALYSIS OF THE BASIC FAILURE EVENTS

To investigate the basic failure events in the FTs, BBNs can be used, following HCL modelling technique. For instance, the Phoenix HRA Methodology uses HCL and combines BBNs with the FTs to model the influence of Performance Influencing Factors (PIFs) on the failure modes. Examples of such applications can be found in (Ramos et al., 2017, 2016; Shen et al., 2010). (Groth et al., 2019) presents a hybrid algorithm with a set of causal factors, human–machine team tasks and events, using BBN causal models, and Bayesian parameter updating methods. Causes for human error in the context of autonomous ships operations have been explored by some authors, such as, (Man et al., 2015; Porathe et al., 2014; Ramos et al., 2018), and can be further modeled as PIFs using a BBN.

Each HRA method has a different set of PIFs, and most of those have been developed in the context of NPP operations (Ramos et al., 2017). Their applicability in the context of autonomous ships operation must thus be assessed. (Ramos et al., 2018) provide an initial analysis of factors that have been explored in the autonomous ships' operation literature. For instance, for the basic failure event *Reading Error*, the following factors could be modelled as the PIFs below, and the influence of these PIFs on the basic failure events can be modelled through BBNs, following the HCL modeling approach:

i) The Display quality is not sufficiently good, e.g., the alerts do not have different colors and / or the size of the font is too small. The operator could thus not identify the reason for the alert as a collision scenario;
ii) The operator is fatigued and does not read the information with sufficient attention;
iii) The operator is stressed due to the alert and perceived the time available for the task as insufficient, therefore reads the information too fast and misreads it.

A crucial aspect of autonomous ships are the failure events as consequence of software failures. A recent method has been proposed by (Thieme et al., 2020a, 2020b) to identify software failures from a functional point of view and implement the resulting failure events in risk analysis. Failure modes are identified for the desired level of functional analysis of the software. The failure modes are propagated to the output of the software. This output then represents the failure modes that are used in the generic fault tress presented in this paper. The functional view that underlies the method (Thieme et al., 2020a, 2020b) makes it possible to analyze possible software failures from an early

stage when the software is specified. The method is programming language independent and therefore even conceptual system may be analyzed.

Additional possibilities of analysis can be explored for use in combination with the H-SIA FTs include a method for analysis of interdependencies in infrastructure systems introduced by (Utne et al., 2011); a method for software analysis in combination with hardware failures (Mutha et al., 2013); the context specific risk management method employed by NASA to analyze system risk (Guarro et al., 2013); HRA or human factors methods for exploring the human failure modes (Endsley, 2017; Kaber, 2018)

4. CASE STUDY

This section introduces the application of the extended H-SIA method to a case study. The scenario is an autonomous ship with low LoA (remote control) and is intentionally kept simple to demonstrate the use of the generic elements.

The autonomous ship (AS) is in a collision situation with a small boat. The AS is equipped with all necessary sensing capabilities to detect objects and other ships or boats, and to estimate their course. Upon detection of an object on collision course the ship will raise an alarm in the SCC (Figure 17). The SCC operators will then decide on a course and send the command to the ship. The autonomous ship should then implement the course. The ship can communicate through the mobile network (5^{th} generation network, 5G) or via satellite link with the SCC. It is assumed that in this case the ship is close enough to shore to use the mobile 5G network. To perform a conservative analysis, it is assumed that the AS cannot communicate with the small boat and the boat will not take sufficient action, i.e., the AS the sole responsible for avoiding collision.

The AS uses Radar and camera images for detection of obstacles and collision candidates. The information is processed on board. In addition to the alert sent to the operator upon any detection of obstacles, the ship also sends the video feed and the radar image to the SCC. The ship is equipped with conventional propeller, ruder and one bow thruster. The SCC is equipped with appropriated displays, in which the operator has access to data on the ship and its surrounding, and the visual and sonorous alerts. In case the ship fails in detecting the collision candidate (CC), the operator can visually detect it in the HMI displays.

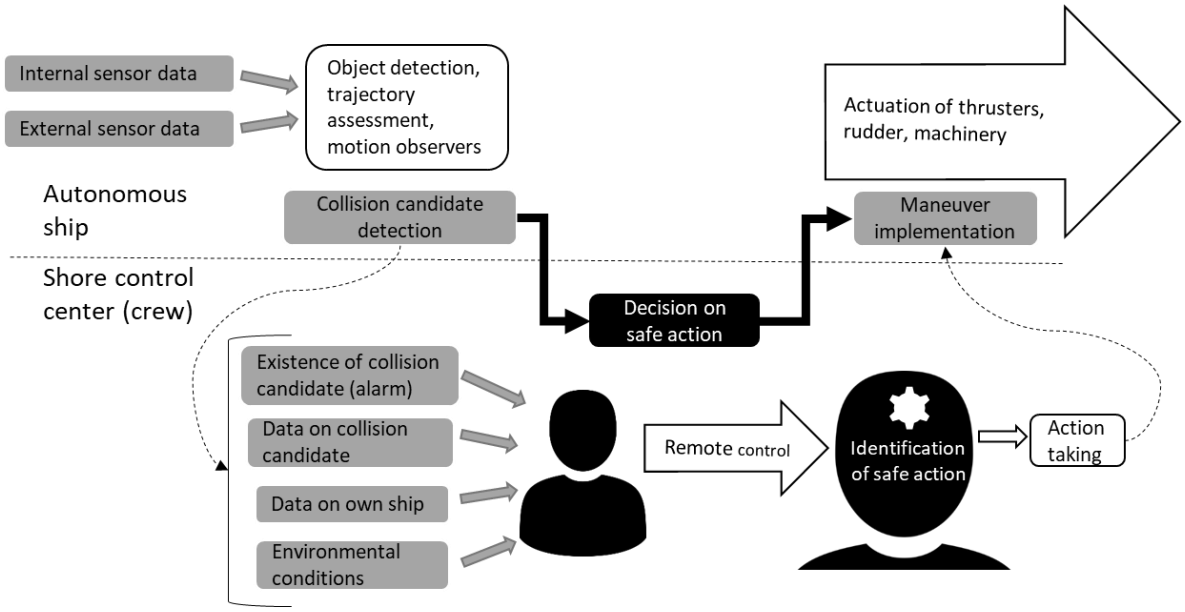

FIGURE 17: CASE STUDY FOR EXTENDED H-SIA APPLICATION

The application of H-SIA consists of the development of the ESD (section 4.1), the analysis of the events of the ESD in a success-oriented manner using the CoTA (section 4.2), and the analysis of these events in a failure-oriented manner, using the FTs (section 4.3).

4.1. EVENT SEQUENCE DIAGRAM

The ESD of the scenario (Figure 18), containing events of both the operators and the autonomous ship, was built following the flowchart in (Ramos et al., 2020). The initiating event (IE) concerns the ship being on collision course.

Event	Description	Agent	Task in the CoTA
A	Detection by autonomous ship	AS	S1
B	Detection by shore control center	SCC	H1
D	Response by operator to alarm in the shore control center	SCC	H2
F	Remote control by shore control center operator	SCC	H3
G	Implementation and execution of collision avoidance plan	AS	H2
//	Parallel tasks		SP T1 SP T2 H5

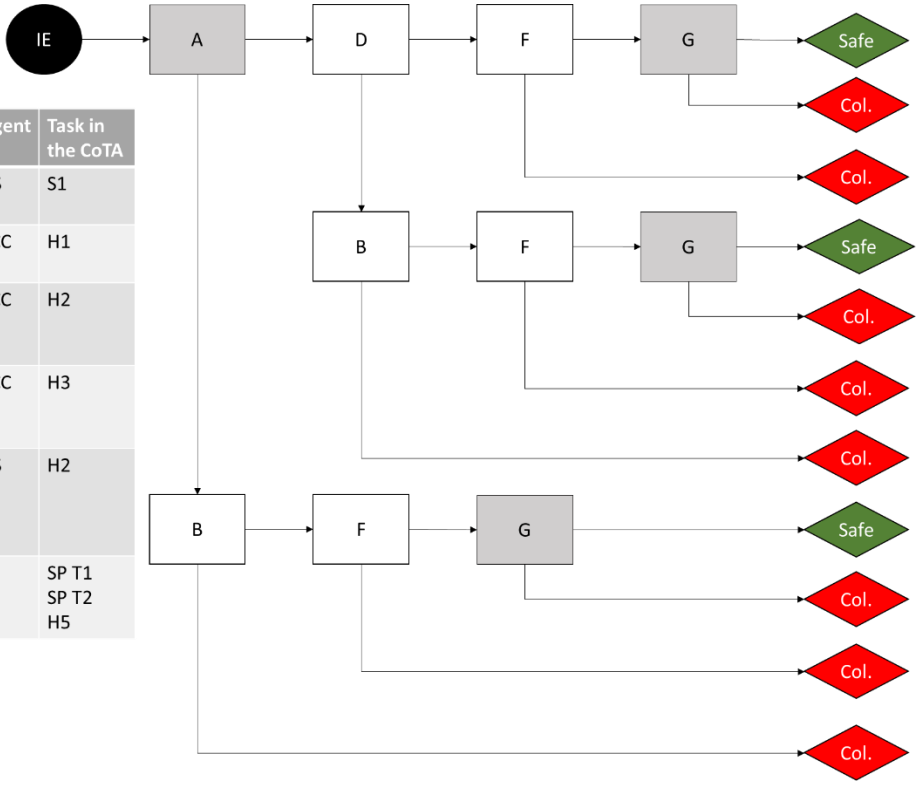

FIGURE 18 ESD DEVELOPED FOLLOWING THE ESD FLOWCHART

4.2. CONCURRENT TASK ANALYSIS

Based on the description of the system, a CoTA is conducted for the autonomous ship and the operator in the SCC (Figure 19 and Figure 20). The development of the CoTA follows the rules established in (Ramos et al., 2020), summarized below as:

1. Definition of agents to be analyzed and definition of Task 0 (avoid collision);
2. Define agents acting in the events and high-level tasks. Each event of the ESD will translate into a high-level task in each of the respective Hierarchical Task Analysis;
3. Identify parallel tasks;
4. Re-describe tasks until reaching stop-rule: the sub-tasks must be associated with only one of the IDA phases and, the interface tasks must be clearly defined.

4.3. FAULT TREES

The FTs are developed for each of the failure events in the ESD, using the applicable paths of the generic FTs. Two FTs are presented in this section: the FT of the Event A *Detection by autonomous ship* and the following event in the success path, event D *Response by operator to alarm in the shore control center*. The FTs are truncated for space optimization, showing only the relevant branches for the events in the case study. The description of the events has been altered and extended to reflect the particulars of the case study. The non-developed branches are signaled with a "(…)".

Note that in this case study the ship has two parallel tasks: (a) Collect information and (b) Listen and answer to commands and requests. The failure in these tasks will be modeled through their own FTs. For example, the interface tasks concerning ship tasks that receive an input from the SCC all concern

action tasks from the humans: sending commands for more information or for remote control. Human Failure in Action will, thus, be modeled through its own FT in order to be connected to the FTs of the ship through a gate.

4.3.1 CASE STUDY FAULT TREE EXAMPLE FOR THE AUTONOMOUS SHIP

Event A covers the detection of the obstacle by the autonomous ship. The related FTs are presented in the following Figures (Figure 21 , Figure 22, and Figure 23). Note that, as indicated in the CoTA, the event of detection of the CC comprises also the notification of the operator. A failure in this event may happen due to a failure in collecting the necessary data, a failure in making the correct situation assessment given the collected data, or a failure in notifying the operator about the detection. Thus, the FT includes all IDA phases. The relevant sensors and system have been implemented on a high level, since the particulars of the system are not fully defined (Figure 22 and Figure 23). Camera and Radar failure will result in the obstacle not being identified, similarly if the motion (speed and heading) is not assessed or wrongly assessed, this will lead to a collision with the obstacle.

If the data is assessed correctly, the obstacle may be not recognized as such or the ships heading is not assessed correctly. The ship may also fail to notify the operator, which will be the follow up action of the ship on a detected obstacle.

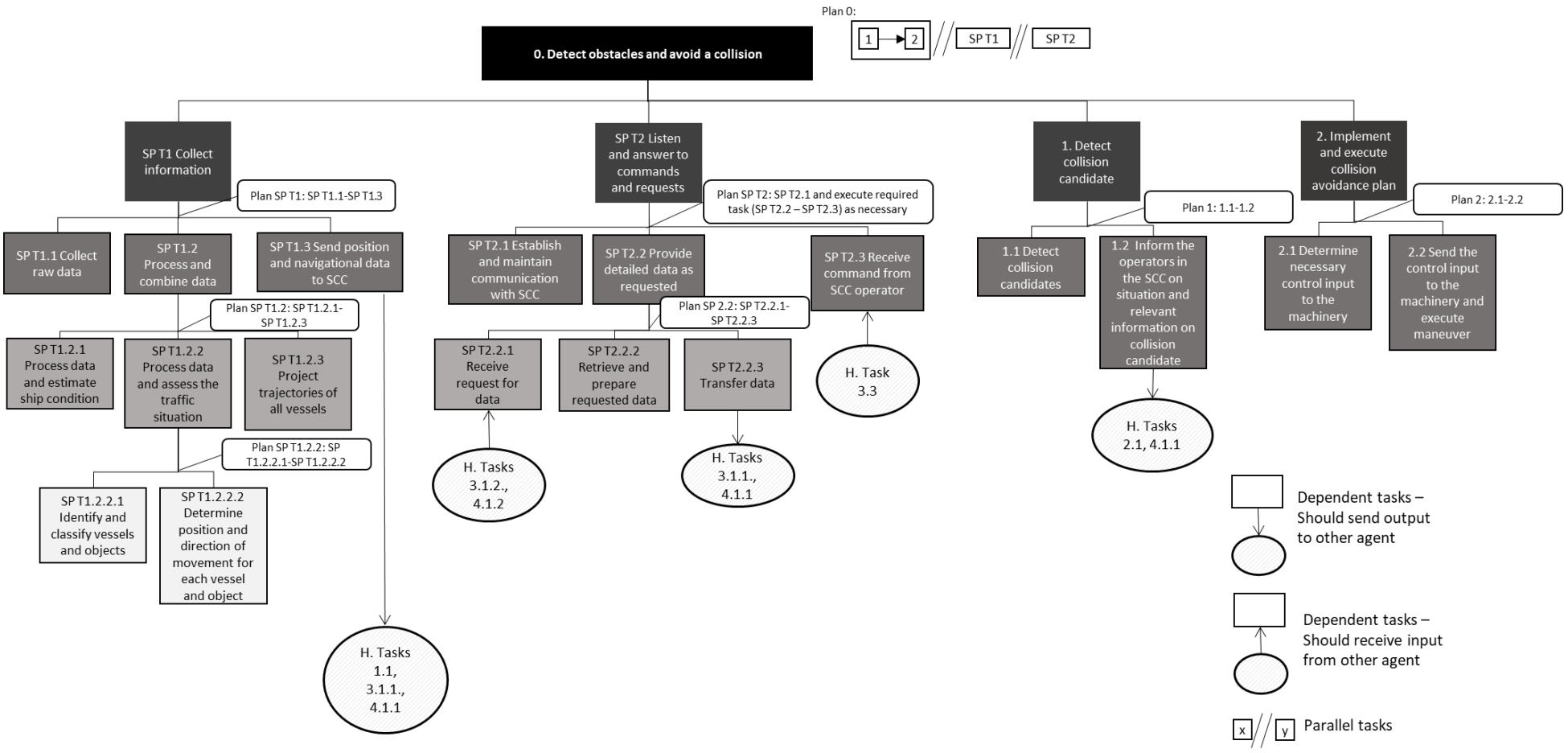

FIGURE 19 COTA FOR THE AUTONOMOUS SHIP

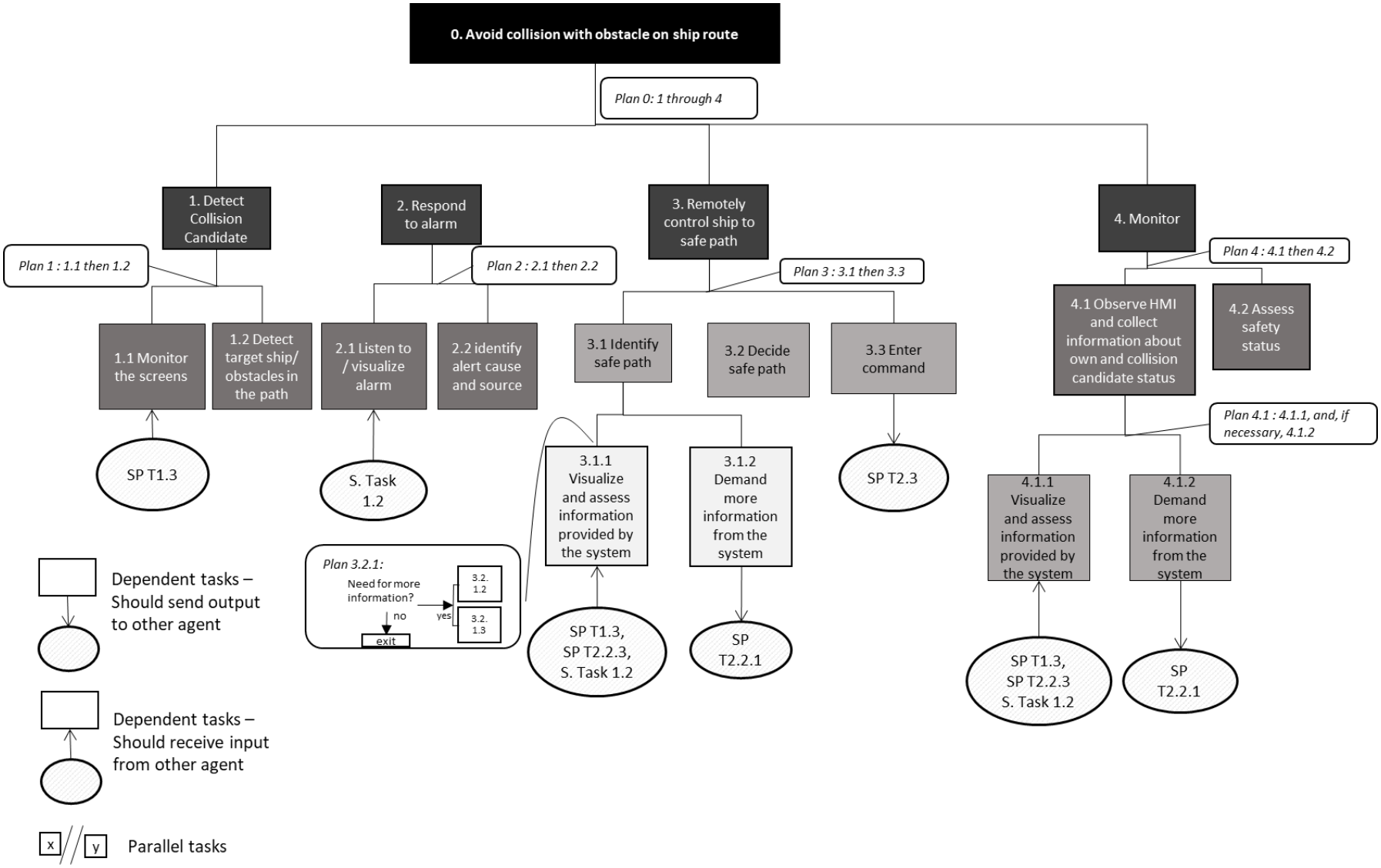

FIGURE 20 COTA FOR THE HUMAN OPERATOR OF THE AUTONOMOUS SHIP

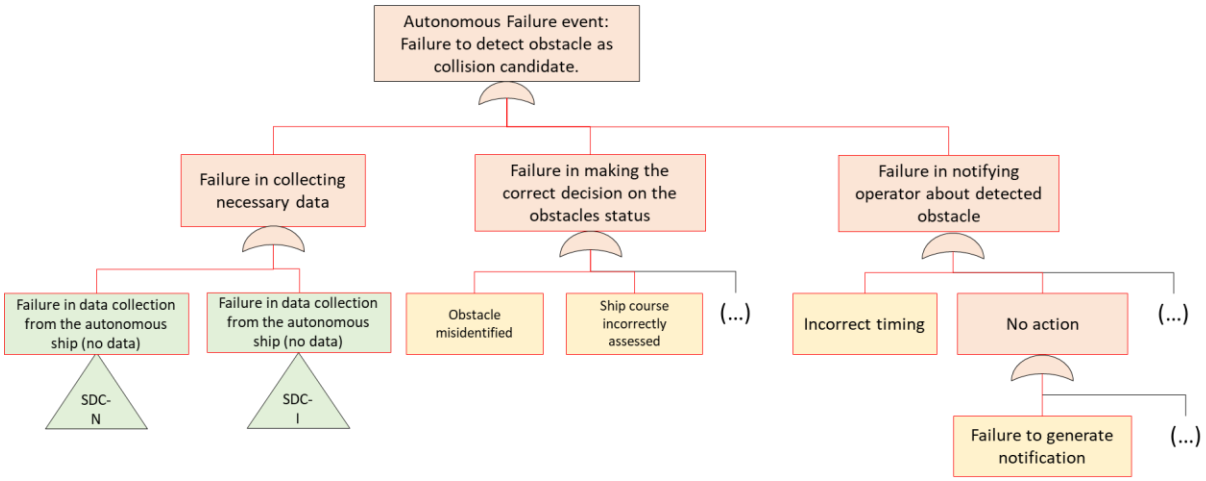

FIGURE 21: FAULT TREE FOR THE EVENT A – DETECTION BY AUTONOMOUS SHIP: FAILURE TO DETECT ALLISION OBSTACLE AS COLLISION CANDIDATE.

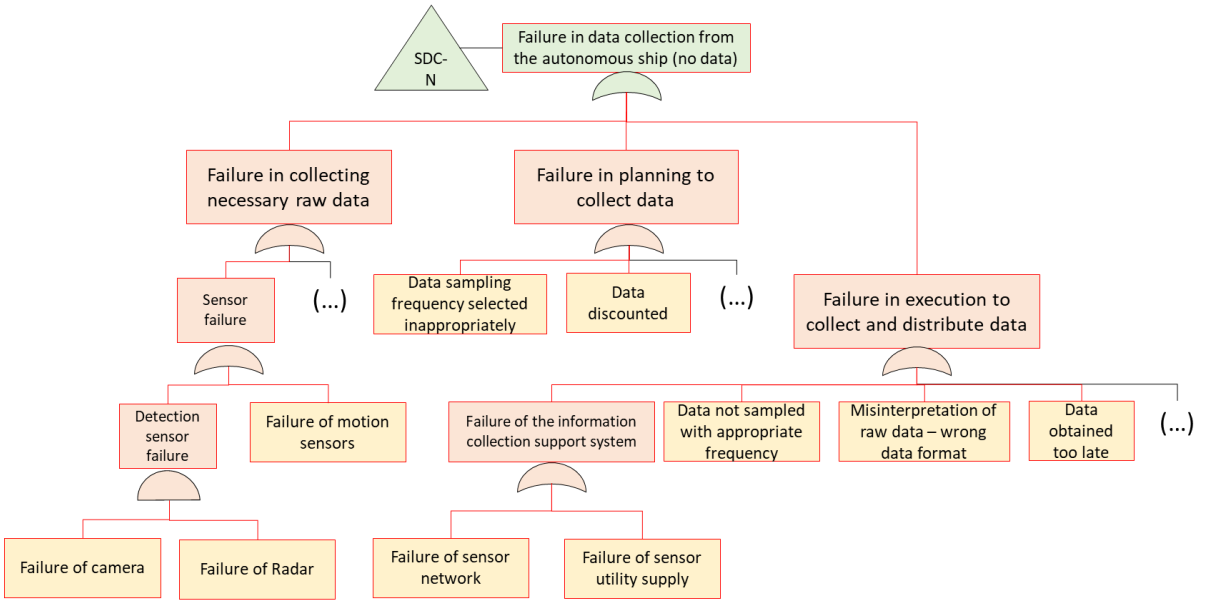

FIGURE 22: FAULT TREE FOR THE SHIP PARALLEL TASK 1 NO DATA FOR EVENT A – DETECTION BY AUTONOMOUS SHIP

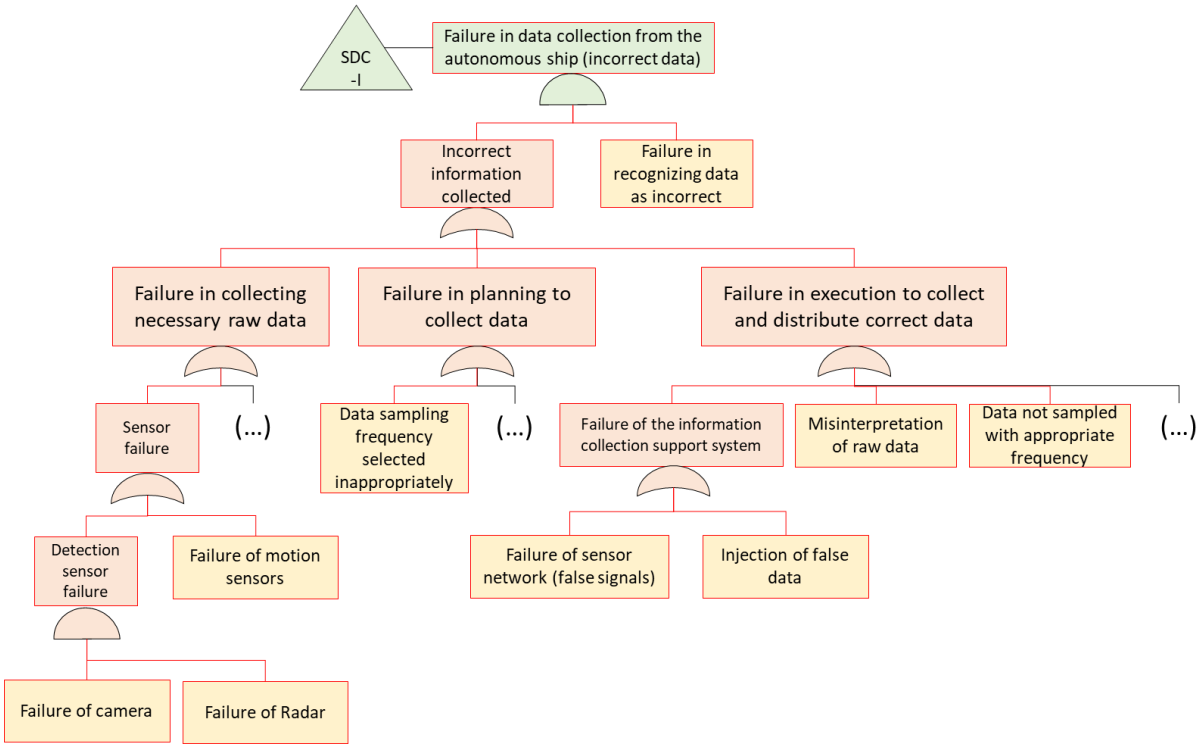

FIGURE 23 FAULT TREE FOR THE SHIP PARALLEL TASK 1 INCORRECT DATA FOR EVENT A – DETECTION BY AUTONOMOUS SHIP

Basic failure events for the decision process (D-phase) could be assessed in more detail through some of the software risk assessment methods as described in Section 3.5, e.g., (Mutha et al., 2013; Thieme et al., 2020a, 2020b). The decision making in autonomous ships is based on software. The mentioned methods can assist in finding relevant basic failure events that can be reflected in the FTs. These methods model the software behavior and analyze the possible behavior in case of software internal failures or hardware failures that influence software behavior.

4.3.2 CASE STUDY FAULT TREE EXAMPLE FOR THE SCC OPERATOR

Event D concerns the operator responding to the alert sent by the ship. Note that it comprises not only the visualization of the alarm, but also the understanding of its source, as can be seen in the CoTA. In other words, the operator must visualize the alarm and understand that it concerns a potential collision, and the identification of the possible collision object. The operator may thus fail in this event if s/he does not collect necessary data, and if s/he does not have a correct situation assessment. Moreover, the operator will fail if there is no communication established between the SCC and the ship, in which case s/he will not receive the alert (Figure 24).

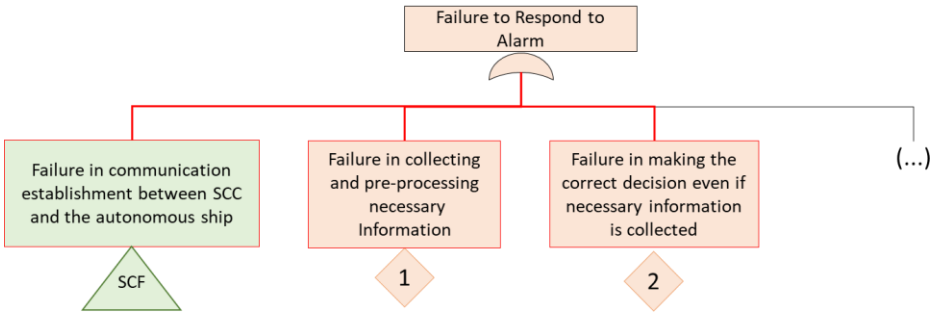

FIGURE 24: FAULT TREE FOR HUMAN FAILURE TO RESPOND TO ALARM: TOP EVENT

The operator may fail in collecting information for alarm response if s/he does not receive information from the ship, due to a failure in data collection or due to a failure in the HMI. The HMI may also display incorrect information, and the operator may fail in recognizing it as incorrect. Moreover, s/he may unintentionally or intentionally not respond to the alarm. After visualizing the alarm, s/he may fail to understand its cause due to a reading error in the display, or for misunderstanding the data (Figure 25, Figure 26 and Figure 27). S/he may also misdiagnose the situation, while trying to understand the causes of the alarm (Figure 28).

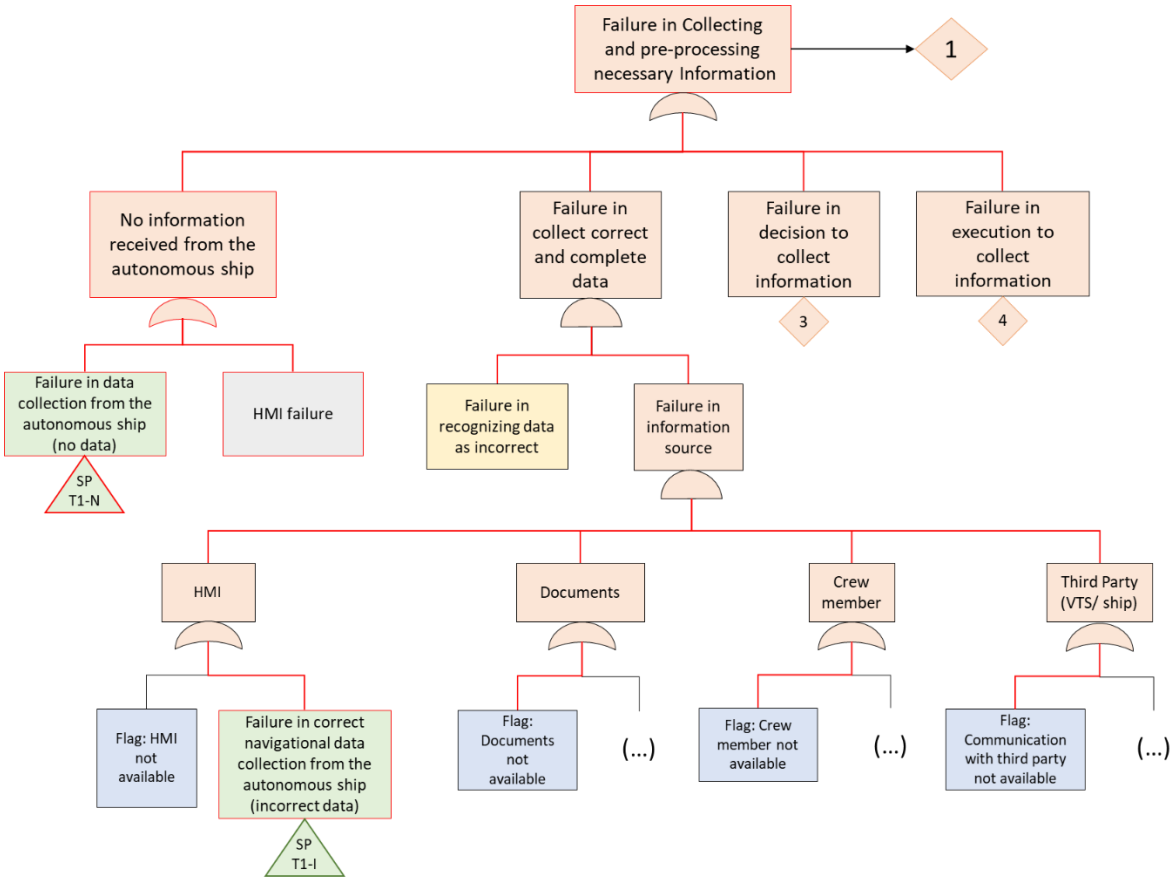

FIGURE 25: FAULT TREE FOR HUMAN FAILURE TO RESPOND TO ALARM: FAILURE IN COLLECTING AND PRE-PROCESSING INFORMATION

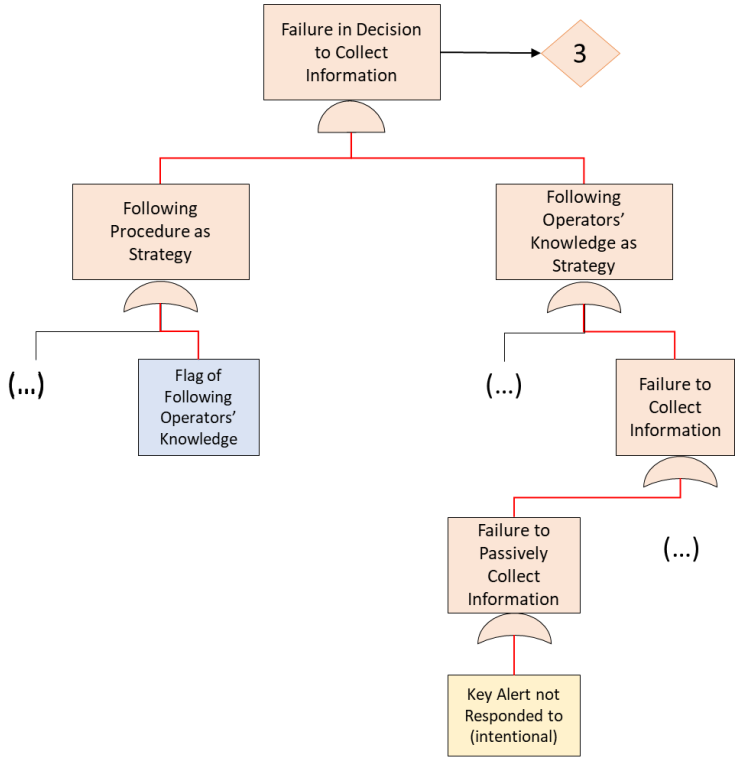

FIGURE 26 FAULT TREE FOR HUMAN FAILURE TO RESPOND TO ALARM: FAILURE IN DECISION TO COLLECT INFORMATION.

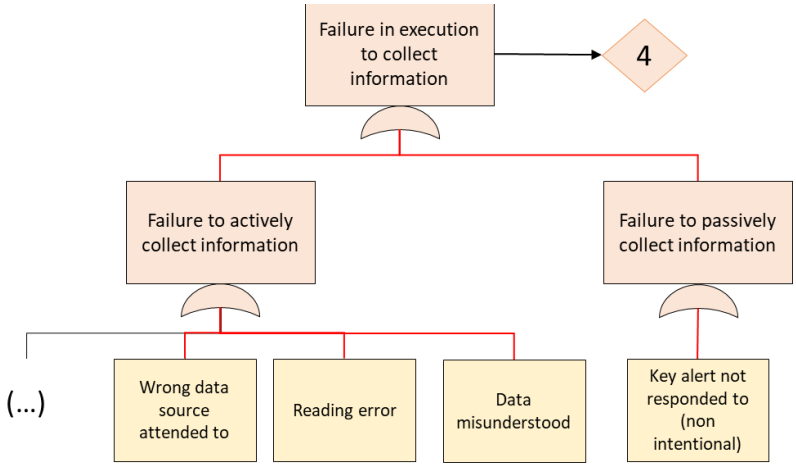

FIGURE 27: FAULT TREE FOR HUMAN FAILURE TO RESPOND TO ALARM: FAILURE IN EXECUTION TO COLLECT INFORMATION.

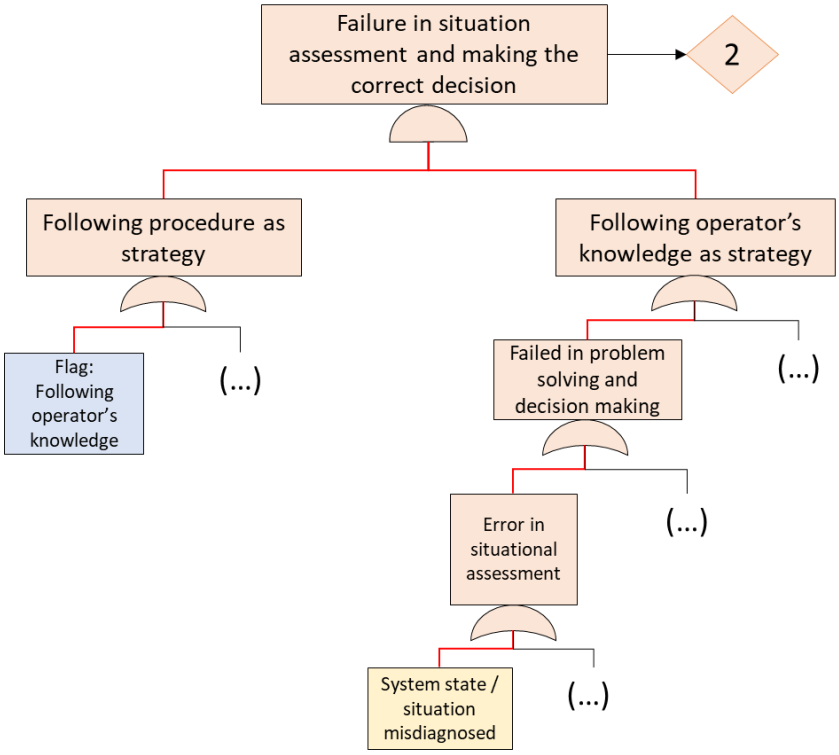

FIGURE 28: FAULT TREE FOR HUMAN FAILURE TO RESPOND TO ALARM: FAILURE IN SITUATION ASSESSMENT AND DECISION-MAKING.

The basic failure events for the operators' failure in responding to the alarm are thus:

i) Failure in recognizing data as incorrect;
ii) Key alert not responded to;
iii) Wrong data source attended to;
iv) Reading error;
v) Data misunderstood;
vi) System state misdiagnose.

4.4 DISCUSSION

An important overall outcome of using the FTs developed using H-SIA is that they may lead to cut sets that would be otherwise not included in the analysis. For instance, when analyzing the operator's failure to respond to an alarm using only Human Reliability Analysis methods, the cut sets would generally not include basic failure events concerning the system's failure in data collection or communication. As illustrated by the case study, these basic events would be identified in a cut set when using H-SIA. Similarly, when analyzing the system's failure in implementing a command, a software analysis method may not include the basic event of receiving an incorrect command from the operator. While these failures do not occur within the envelope of operation of the system or human strictly speaking (if those were analyzed separately), they do propagate from one to another agent of the system.

In comparison to advanced hazard analysis methods, the H-SIA extension with fault trees leads directly to quantifiable risk models. This is an advantage over other methods, since risk analysts are familiar

with fault tree analysis and the subsequent interpretation of results. Similar to FRAM and STPA, the presented method is scalable to the level of analysis. For example, a detailed design may require a higher level of failure analysis in comparison to a preliminary design. H-SIA's fault trees can accommodate these details if necessary. A limitation of the H-SIA FTs concerns its structure, which focuses on the functions, and their abstracted sub-functions. Analysts may prefer a component and sub-system view. However, focusing only on components and their failures may overlook significant interactions (Leveson, 2012).

In comparison to the existing risk models for autonomous ships (Jensen, 2015; Rødseth and Burmeister, 2015) the analysis presented in this article is more detailed and provides more in-depth explanation of failure causes and events. For example, Jensen (2015) focus is on the identification possible scenarios that lead to being on a collision course. However, hardware failure is the focus of these models. According to (Thieme et al., 2018) no current quantitative risk models for marine vessels are including software and human operator interaction sufficiently. The chosen case-study example is rather brief to demonstrate the concept of the H-SIA extension. A full analysis for an autonomous ship will be even more detailed in order to provide a sound risk analysis.

5. CONCLUDING REMARKS

Autonomous systems present a high level of complexity concerning the interactions between software, hardware and humans through dynamic LoA. Recent studies have approached this issue in different manners, e.g. (Kari et al., 2018; Wróbel et al., 2018a). These studies have focused mainly on hazard identification, which is an important input for risk assessment. Nonetheless, in general, these studies employ approaches that are not quantifiable. Moreover, despite the important focus on the interactions between the sub-systems, some studies do not provide the same level of detail of analysis to behaviors that occur within a sub-system. H-SIA aims at modeling behaviors and failure that occur in the interactions between the sub-systems as well as within each sub-system. For instance, the human operator is modeled not only at the level of giving commands and receiving information, but also at the level of diagnosis and situation assessment. This is possible due to the use of the IDA model for the operator and for the autonomous system. Indeed, an important feature of H-SIA concerns the explicit inclusion and analysis of human failures. Humans will be involved in autonomous systems operation in the near and medium future, risk methods must thus consider their failures. H-SIA provides, thus, additional layers of analysis than hazard identification. H-SIA and other approaches used in previous studies can be used for different purposes of analysis and can possibly complement themselves.

The H-SIA method comprised initially ESD and a CoTA. The extension of the method presented in this paper included a failure-orientation analysis, using Fault Trees. A case study shows that H-SIA can be used and aid with risk assessments of these systems. The use of the suggested methods for analyzing the causes of the basic failure events can provide a causal model that can be used for the identification of risk reduction measures.

We provide guidelines for developing generic FTs using the CoTA. The FTs allow the analysis of the hardware, software, and human failures in the context of a scenario. The FTs in H-SIA acknowledge that the autonomous system is composed of interacting sub-systems or agents. The failure in one task

of one agent can lead to a failure of another agent's task, in case there are dependencies between the tasks. The use of FTs developed using H-SIA can lead to a cut sets that would be otherwise not included in the analysis, as illustrated by the case study.

A valuable feature of H-SIA's generic FTs is the basic failure events. The latter can be used to explain how the system can fail and are generic for use in different systems and designs. The basic failure events may need further analyses to reflect the failure mechanisms or their failure causes acting on these system elements in detail. Although this paper outlines some of the possible approaches for further analysis of the failure causes, H-SIA does not yet provide a guideline for optimal choice of the method. This guideline is subject of future work.

Moreover, it should be noted that H-SIA applies static methods, such as FTs, whereas autonomous systems are operated dynamically. The extended H-SIA method may be used for design risk analysis, where the dynamics are not of much concern, but also during operation – given that the analyst judges that a static method can model the operation with sufficient accuracy as needed for the analysis. The H-SIA framework is, so far, qualitative. Future work includes a quantitative framework, in which probabilities and uncertainties can be updated during operation and interfaced with online measurements of the system's condition. Here H-SIA can support the development of such a system through the generic risk models in the ESD and FTs, as well as the CoTA. These elements can help to identify risk indicators covering the success and the failure side of operation.

The method presented in this paper still needs to be validated using experiments or simulation methods. Since autonomous ships are not yet in operation phase, real full-scale experiments are challenging. However, it is possible to use simulations for testing the method. Since the main feature of the method is modeling the interactions between sub-systems, as well as the behavior of each sub-system (e.g. human operator and software), a simulation platform that includes this interaction would be suitable. A possible candidate is the approach used in the Accident Dynamic Simulator with the IDAC model (ADS-IDAC)(Chang and Mosleh, 2007b). ASD-IDAC models the physical parameters of the system, external conditions, and the cognitive phases of the operators. It was developed for Nuclear Power Plant operations and could be expanded for autonomous ships operations. An additional possibility concerns model experiments / testing. Typically, with marine structures (such as ships or platforms), simulations are performed, then model testing is executed in a lab basin/facility and then real experiments/testing can be performed during and after commissioning. Both simulations and model experiments / testing are subject of future work.

ACKNOWLEDGMENTS

This work has been partly supported by the Research Council of Norway through the Centre of Excellence funding scheme, NTNU AMOS, project number 223254, and the UNLOCK project, through the Research Council of Norway FRINATEK scheme, project number 274441.

REFERENCES

Annett, J., Stanton, N., 2000. Research and developments in task analysis, in: Annett, J., A.Stanton, N. (Eds.), Task Analysis. Taylor & Francis, London, pp. 1–8.

Chang, Y.H.J., Mosleh, A., 2007a. Cognitive modeling and dynamic probabilistic simulation of operating

crew response to complex system accidents . Part 2 : IDAC performance influencing factors model 92, 1014–1040. https://doi.org/10.1016/j.ress.2006.05.010

Chang, Y.H.J., Mosleh, A., 2007b. Cognitive modeling and dynamic probabilistic simulation of operating crew response to complex system accidents Part 1 : Overview of the IDAC Model 92, 997–1013. https://doi.org/10.1016/j.ress.2006.05.014

Di, Z., Mingyang, Z., Houjie, Y., Kai, Z., Cunlong, F., 2020. Prediction Model of Human Error Probability in Autonomous Cargo Ships. Proc. Int. Semin. Saf. Secur. Auton. Vessel. Eur. STAMP Work. Conf. 2019 110–124. https://doi.org/10.2478/9788395669606-010

Diaper, D., Stanton, N., 1992. The Handbook of Task Analysis for Human-Computer Interaction.

Ekanem, N.J., Mosleh, A., Shen, S.-H., 2015. Phoenix–A model-based Human reliability analysis methodology: Qualitative analysis procedure. Reliab. Eng. Syst. Saf. 145, 1–15. https://doi.org/10.1016/j.ress.2015.07.009

Endsley, M.R., 2017. From Here to Autonomy: Lessons Learned from Human-Automation Research. Hum. Factors 59, 5–27. https://doi.org/10.1177/0018720816681350

Fan, C., Wróbel, K., Montewka, J., Gil, M., Wan, C., Zhang, D., 2020. A framework to identify factors influencing navigational risk for Maritime Autonomous Surface Ships. Ocean Eng. 202, 107188. https://doi.org/10.1016/j.oceaneng.2020.107188

Groth, K., Wang, C., Mosleh, A., 2010. Hybrid causal methodology and software platform for probabilistic risk assessment and safety monitoring of socio-technical systems. Reliab. Eng. Syst. Saf. 95, 1276–1285. https://doi.org/10.1016/j.ress.2010.06.005

Groth, K.M., Smith, R., Moradi, R., 2019. A hybrid algorithm for developing third generation HRA methods using simulator data, causal models, and cognitive science. Reliab. Eng. Syst. Saf. 191, 106507. https://doi.org/10.1016/j.ress.2019.106507

Groth, K.M., Wang, C., Zhu, D., Mosleh, A., 2009. Methodology and software platform for multi-layer causal modeling. Safety, Reliab. Risk Anal. Theory, Methods Appl. - Proc. Jt. ESREL SRA-Europe Conf. 1, 113–120.

Guarro, S.B., Yau, M.K., Dixon, S., 2013. Context-Based Software Risk Model (CSRM) Application Guide. ASCA Inc., Washington, D.C. 20546.

Hendrickson, S., Whaley, A., Boring, R., Shen, S.-H., Mosleh, A., Oxstrand, J., Forester, J., Kelly, D., 2010. A Mid-Layer Model for Human Reliability Analysis: Understanding the Cognitive Causes of Human Failure Events, in: International Conference on Probabilistic Safety Assessment and Management PSAM 10. Washington.

Ho, G., Pavlovic, N., Arrabito, R., 2011. Human factors issues with operating unmanned underwater vehicles. Proc. Hum. Factors Ergon. Soc. 55, 429–433. https://doi.org/10.1177/1071181311551088

Hogenboom, S., Rokseth, B., Vinnem, J.E., Utne, I.B., 2020. Human reliability and the impact of control function allocation in the design of dynamic positioning systems. Reliab. Eng. Syst. Saf. 194. https://doi.org/10.1016/j.ress.2018.12.019

Hollnagel, E., 2017. FRAM: The Functional Resonance Analysis Method, 1st Ed. ed, FRAM: The Functional Resonance Analysis Method. Ashgate, Farnham. UK. https://doi.org/10.1201/9781315255071

Huang, H.-M., Messina, E., Jacoff, A., 2010. Performance Measures Framework for Unmanned Systems(PerMFUS) - Initial Perspective. PerMIS'09. https://doi.org/ACM 978-1-60558-747-

9/09/09

IEC, 2010. IEC 61508: Functional safety of electrical/electronic/ programmable electronic safety related systems.

International Maritime Organisation, 1972. COLREGs - International Regulations for Preventing Collisions at Sea, International Maritime Organization -Convention on the International Regulations for Preventing Collisions at Sea, 1972.

Jensen, F., 2015. Hazard and Risk Assessment of Unmanned Dry Bulk Carriers on the High Seas. Technische Universität Hamburg.

Kaber, D.B., 2018. A conceptual framework of autonomous and automated agents. Theor. Issues Ergon. Sci. 19, 406–430. https://doi.org/10.1080/1463922X.2017.1363314

Kari, R., Gaspar, H.M., Gausdal, A.H., Morshedi, M., 2018. Human Interactions Framework for Remote Ship Operations. MED 2018 - 26th Mediterr. Conf. Control Autom. 581–587. https://doi.org/10.1109/MED.2018.8442624

Leveson, N.G., 2012. Engineering a safer world: systems thinking applied to safety. Choice Rev. Online. https://doi.org/10.5860/choice.49-6305

Leveson, N.G., Fleming, C.H., Spencer, M., Thomas, J., Wilkinson, C., 2012. Safety Assessment of Complex, Software-Intensive Systems. SAE Int. J. Aerosp. 5, 233–244. https://doi.org/10.4271/2012-01-2134

Leveson, N.G., Thomas, J.P., 2018. STPA Handbook, 1. ed. Cambridge, MA, USA.

Li, S., Meng, Q., Qu, X., 2012. An Overview of Maritime Waterway Quantitative Risk Assessment Models. Risk Anal. 32, 496–512. https://doi.org/10.1111/j.1539-6924.2011.01697.x

Man, Y., Lundh, M., Porathe, T., Mackinnon, S., 2015. From desk to field - Human factor issues in remote monitoring and controlling of autonomous unmanned vessels. Procedia Manuf. 3, 2674–2681. https://doi.org/10.1016/j.promfg.2015.07.635

Mosleh, A., 2014. PRA: A Perspective on strengths, current Limitations, and possible improvements. Nucl. Eng. Technol. 46, 1–10. https://doi.org/10.5516/NET.03.2014.700

Mosleh, A., Forester, J., Boring, R., Hendrickson, S., Whaley, A., Shen, S.-H., Kelly, D., Chang, J., Dang, V., Oxstrand, J., Lois, E., 2010. A model-based human reliability analysis framework, in: International Conference on Probabilistic Safety Assessment and Management PSAM 10. Washington.

MUNIN, 2016. Research in maritime autonomous systems project results and technology potentials.

Mutha, C., Jensen, D., Tumer, I., Smidts, C., 2013. An integrated multidomain functional failure and propagation analysis approach for safe system design. Artif. Intell. Eng. Des. Anal. Manuf. 27, 317–347. https://doi.org/10.1017/S0890060413000152

Otto, S., Pedersen, P.T., Samuelides, M., Sames, P.C., 2002. Elements of risk analysis for collision and grounding of a RoRo passenger ferry. Mar. Struct. 15, 461–474. https://doi.org/http://dx.doi.org/10.1016/S0951-8339(02)00014-X

Ozarin, N.W., 2013. Bridging software and hardware FMEA in complex systems, in: 2013 Proceedings Annual Reliability and Maintainability Symposium (RAMS). pp. 1–6. https://doi.org/10.1109/RAMS.2013.6517739

Ozarin, N.W., 2009. Applying software failure modes and effects analysis to interfaces, in: Annual

Reliability and Maintainability Symposium . pp. 533–538. https://doi.org/10.1109/RAMS.2009.4914732

Patriarca, R., Bergström, J., Di Gravio, G., 2017. Defining the functional resonance analysis space: Combining Abstraction Hierarchy and FRAM. Reliab. Eng. Syst. Saf. 165, 34–46. https://doi.org/10.1016/j.ress.2017.03.032

Peter Barthelsson, J.S., Sagefjord, J., 2017. Autonomous ships and the operator's role in a Shore Control Centre - A comparative analysis on projects in the Scandinavian region and implementing the experience of Mariners to a new field of shipping.

Porathe, T., 2014. Remote Monitoring and Control of Unmanned Vessels – The MUNIN Shore Control Centre, in: Proceedings of the 13th International Conference on Computer and IT Applications in the Maritime Industries COMPIT'14. Redworth, pp. 460–467.

Porathe, T., Prison, J., Man, Y., 2014. SITUATION AWARENESS IN REMOTE CONTROL CENTRES FOR UNMANNED SHIPS, in: Proceedings of the Human Factors in Ship Design & Operation Conference. London.

Ramos, M.A., Droguett, E.L., Mosleh, A., 2016. Human Reliability Analysis of an Oil Refinery Operation Using the Phoenix HRA Methodology: A Hydrogen Generation Unit Case STUDY, in: Proceedings of the 13th International Conference on Probabilistic Safety Assessment and Management (PSAM 13). Seoul, pp. 1–8.

Ramos, M.A., Droguett, E.L., Mosleh, A., das Chagas Moura, M., Ramos Martins, M., 2017. Revisiting past refinery accidents from a human reliability analysis perspective: The BP Texas City and the Chevron Richmond accidents. Can. J. Chem. Eng. 95, 2293–2305. https://doi.org/10.1002/cjce.22996

Ramos, M.A., Thieme, C.A., Utne, I.B., Mosleh, A., 2020. Human-system concurrent task analysis for maritime autonomous surface ship operation and safety. Reliab. Eng. Syst. Saf. 195, 106697. https://doi.org/10.1016/j.ress.2019.106697

Ramos, M.A., Utne, I.B., Mosleh, A., 2019. Collision avoidance on maritime autonomous surface ships: Operators' tasks and human failure events. Saf. Sci. 116, 33–44. https://doi.org/10.1016/j.ssci.2019.02.038

Ramos, M.A., Utne, I.B., Mosleh, A., 2018. On factors affecting autonomous ships operators performance in a Shore Control Center, in: Proceedings to the Probabilistic Safety Assessment and Management PSAM 14. Los Angeles.

Rausand, M., 2011. Risk Assessment - Theory, Methods, and Applications, 1st Ed. ed. John Wiley & Sons, Hoboken, New Jersey, USA.

Reason, J., 1990. Human Error. Cambridge University Press. https://doi.org/10.1017/CBO9781139062367

Ristord, L., Esmenjaud, C., 2002. FMEA Performed on the SPINLINE3 Operational System Software as part of the TIHANGE 1 NIS Refurbishment Safety Case. Cnra/Csni Work. Licens. Oper. Exp. Comput. I&C Syst.

Rødseth, Ø.J., Burmeister, H.-C., 2015. Risk Assessment for an Unmanned Merchant Ship. TransNav, Int. J. Mar. Navig. Saf. Sea Transp. 9, 357–364. https://doi.org/10.12716/1001.09.03.08

Rødseth, Ø.J., Nordahl, H., 2017. Definitions for Autonomous Merchant Ships", Norwegian Forum for Unmanned Ships, Version 1.0, Norwegian Forum for Unmanned Ships.

Roelen, A., Wever, R., Mosleh, A., Groth, K., 2008. Development and validation of a comprehensive

hybrid causal model for safety assessment and management of aviation systems. 9th Int. Conf. Probabilistic Saf. Assess. Manag. 2008, PSAM 2008 1, 776–783.

Shen, S.-H., Mosleh, A., Kelly, D., Boring, R., 2010. Example Application of Model-Based HRA Approach, in: International Conference on Probabilistic Safety Assessment and Management PSAM 10. Washington.

Shen, S.-H., Smidts, C., Mosleh, A., 1997. A methodology for collection and analysis of human error data based on a cognitive model: IDA. Nucl. Eng. Des. 172, 157–186. https://doi.org/10.1016/S0029-5493(97)00002-2

Shepherd, A., 2001. Hierarchical Task Analysis. Taylor & Francis, London.

Swain, A.D., Guttmann, H.E., 1983. Handbook of human reliability analysis with emphasis on nuclear power plant applications, NUREG/CR-1278, null.

Swaminathan, S., Smidts, C., 1999a. The Event Sequence Diagram framework for dynamic Probabilistic Risk Assessment. Reliab. Eng. Syst. Saf. 63, 73–90. https://doi.org/https://doi.org/10.1016/S0951-8320(98)00027-1

Swaminathan, S., Smidts, C., 1999b. The mathematical formulation for the event sequence diagram framework. Reliab. Eng. Syst. Saf. 65, 103–118. https://doi.org/DOI: 10.1016/S0951-8320(98)00092-1

Thieme, C.A., Mosleh, A., Utne, I.B., Hegde, J., 2020a. Incorporating software failure in risk analysis – Part 1: Software functional failure mode classification. Reliab. Eng. Syst. Saf. 197, 106803. https://doi.org/10.1016/j.ress.2020.106803

Thieme, C.A., Mosleh, A., Utne, I.B., Hegde, J., 2020b. Incorporating software failure in risk analysis—Part 2: Risk modeling process and case study. Reliab. Eng. Syst. Saf. 198, 106804. https://doi.org/10.1016/j.ress.2020.106804

Thieme, C.A., Utne, I.B., Haugen, S., 2018. Assessing ship risk model applicability to Marine Autonomous Surface Ships. Ocean Eng. 165, 140–154. https://doi.org/10.1016/j.oceaneng.2018.07.040

Utne, I.B., Hokstad, P., Vatn, J., 2011. A method for risk modeling of interdependencies in critical infrastructures. Reliab. Eng. Syst. Saf. 96, 671–678. https://doi.org/10.1016/j.ress.2010.12.006

Utne, I.B., Rokseth, B., Sørensen, A.J., Vinnem, J.E., 2020. Towards supervisory risk control of autonomous ships. Reliab. Eng. Syst. Saf. 196. https://doi.org/10.1016/j.ress.2019.106757

Utne, I.B., Sørensen, A.J., Schjølberg, I., 2017. Risk Management of Autonomous Marine Systems and Operations, in: Proceedings of the International Conference on Offshore Mechanics and Arctic Engineering - OMAE (2017). American Society of Mechanical Engineers (ASME), Trondheim. https://doi.org/10.1115/OMAE2017-61645

Vagia, M., Rødseth, E.J., 2019. A taxonomy for autonomous vehicles for different transportation modes. J. Phys. Conf. Ser. 1357. https://doi.org/10.1088/1742-6596/1357/1/012022

Vagia, M., Transeth, A.A., Fjerdingen, S.A., 2016. A literature review on the levels of automation during the years. What are the different taxonomies that have been proposed? Appl. Ergon. https://doi.org/10.1016/j.apergo.2015.09.013

Wang, C., 2007. Hybrid Causal Logic Methodology for Risk Assessment. University of Maryland.

Wróbel, K., Montewka, J., Kujala, P., 2018a. Towards the development of a system-theoretic model for safety assessment of autonomous merchant vessels. Reliab. Eng. Syst. Saf. 178, 209–224.

https://doi.org/https://doi.org/10.1016/j.ress.2018.05.019

Wróbel, K., Montewka, J., Kujala, P., 2018b. System-theoretic approach to safety of remotely-controlled merchant vessel. Ocean Eng. 152, 334–345. https://doi.org/10.1016/j.oceaneng.2018.01.020

Zou, J., 2018. Systems-Theoretic Process Analysis (STPA) Applied to the Operation of Fully Autonomous Vessels.

APPENDIX A: GENERIC FAILURE EVENTS

TABLE 2: AUTONOMOUS SHIP BASIC FAILURE EVENTS LEADING TO FAILURE IN DATA COLLECTION

Basic Failure Event	Description
Sensor(s) not available or Sensor(s) produce incorrect measurements	One or several sensors fails and either produce no output (SDC-N) or an incorrect output (SDC-I). The sensors involved and the logical connection of these are highly dependent on the situation of concern. Sensor failures may range from optical sensors, to cameras, over radar, or accelerometers.
Failure(s) in databases/ other data servers	Data that is not provided or incorrectly provided may also a failure source for failure of data collection. Similar to the Sensor failure described above, these are highly context specific failures that may have several sub events.
Data sampling frequency selected inappropriately	Depending on the situation, the frequency may be too low, no data is available, since it was note collected yet, or the data was not updated yet. Another example are adaptive sampling algorithms that may decide not to collect information when necessary or too late.
Failure to plan to poll data (*for SDC-N*)	Failure of the system to collect data when this is needed. The data in this case is of sporadic type and not collected continuously. An example of such data could be the weather forecast data that is not polled when an update is available.
Data discounted (*for SDC-N*)	The ship makes the decision to not use available data. This may happen to due to some failure in voting, or erroneous weights.
Incorrect data source selected (*for SDC-I*)	In this case a failure is caused by the system deciding to collect data from the wrong source. Examples and causes may be, weighing of information sources.
Failure of the information collection support system	This is a collective event that may include several basic events that are connected through different logic gates. This basic event refers to these failures in the data collection system that are caused through failed support systems. Events may include failure(s) in the network, loose or broken cables, a (partial) blackout, etc.
Data not sampled with the appropriate frequency	This basic failure event may also cause both failures leading to no data, or failures leading to incorrect data. This FM, in comparison to *Data sampling frequency selected inappropriately* describes the failure of the ship to collect data with the appropriate sampling frequency. Reasons and causes may be found in a network overload or a software-hardware system that cannot process the amount of data fast enough.
Wrong data source attended to	This basic failure event may also lead to no data or incorrect data being collected. Instead of collecting the data from the correct source, another source is used for obtaining that information. That source may not contain the information or have incorrect information. An example may be the retrieval of information from a data server instead of directly from the sensor network.
Misinterpretation of raw data	This basic failure event describes a failure where the obtained raw data is encoded in one data type, but the ship interprets the data as another datatype. This may lead to the case that the information cannot be interpreted (SDC-N), or that the information is transferred into another value and hence incorrect (SDC-I).
Data not obtained (for *SDC-N*)	The autonomous ship polls for information does not obtain the information back. An example may be, if the system asks for a specific

Basic Failure Event	Description
	variable name, but this variable name is different from the variable name, the data is associated with.
Data obtained too late (*SDC-N*)	The system collects the data too late to execute to current action correctly. Similar to the inadequate frequency this may be caused by an overload of the network, or insufficient processing capabilities
Incorrect normalization of data (*SDC-I*)	The data that is supposed to be used is processed incorrectly and hence leads to a failure.
Failure in recognizing data as incorrect (*SDC-I*)	This event summarizes the detection capabilities of a failure during data collection for the event under consideration. Such features may cover only a part of the possible failures described above. Such systems may include validity checks, probabilistic reasoning, etc.

TABLE 3: AUTONOMOUS SHIP BASIC FAILURE EVENTS LEADING TO FAILURE IN COMMUNICATION

Basic Failure Event	Description
Failure(s) of communication equipment	This basic failure event is a set of events connected through logic gates. These failures represent the failure of different communication equipment, e.g., satellites, mobile network, satellite receiver, etc.
Failure to acknowledge communication request	The ship receives a request for communication. However, the ship does not realize the request, e.g., it does not realize that it is the ship called over radio.
Failure to acknowledge data request	The ship does not realize that it is required to send data although the request has been received. This may be caused by a failure to understand the request.
Failure to choose correct communication channel	The ship chose a communication channel that is inadequate for the current situation. E.g., using radio or the mobile network to reach the shore base while being on the high seas.
Decision to delay action	The ship decides to delay further action, due to prioritization of other tasks.
Failure to choose correct communication partner	The ship does not identify the correct communication partner, e.g., it identifies wrongly the calling vessel and informs the operator about the wrong calling vessel.
Failure in recognizing requested data	The ship fails to identify the requested data. This may be due to a request for data with an unknown variable name.
Communication established with the incorrect partner	The ship established the communication with the wrong partner, i.e., it calls upon the wrong vessel through a satellite phone connection.
Incorrect operation of communication equipment	The vessel fails in operating the communication equipment as required. Possible failures include, use of incorrect encryption/ encoding of the information, or use of an inadequate frequency in radio.
Failure of communication equipment	Equipment for communication has a failure and is not operable. This may be caused through software or hardware related failures, such as, failure of antennas, partial blackout, failure of transponders.
Incorrect timing	The ship executes the requested action with respect to communication at the wrong time. In most cases, too late will be the basic failure event.

TABLE 4: AUTONOMOUS SHIP BASIC FAILURE EVENTS LEADING TO FAILURE IN SITUATION ASSESSMENT AND DECISION MAKING

Basic Failure Event	Description
System/ environmental state misdiagnosed	The ship and its algorithms cannot assess the state of the ship and/ or its environment correctly. This may be the position of ship in relation to objects and other ships, or the wave and wind load that may alter the course of the ship.
Failure to adapt strategy to the situation	A strategy planned by the ships algorithms is insufficient for the present situation. A strategy in this article is related to "learned" and adaptive behavior of the system. Examples may be found in a self-learned algorithm for trajectory prediction, or the self-learned collision avoidance strategy that is insufficient in the current situation.
Inappropriate procedure chosen	A procedure followed by the ship is inadequate in the current situation. Procedure in this article refers to directly implemented rules and behaviors in the algorithms of the ship. Examples are turning in the wrong direction, or action if no action is required by the ship.
Decision to delay action	The system may delay further action, e.g., prioritizing other actions.

TABLE 5: AUTONOMOUS SHIP BASIC FAILURE EVENTS LEADING TO THE FAILURE IN ACTION

Basic Failure Event	Description
Action on wrong component	The intended action is not carried out by the intended component. For the ship, this maybe actuation of the wrong thruster. Failure causes for this event may be found in interaction failures, software failure, hardware failure, etc.
Incorrect timing	The intended action is not carried out in the right time. This may be too early, but in most cases a delay will be a relevant failure cause.
Incorrect operation of components	The action is not carried out as expected. This may be too much thrust from the thruster, or too little pitch of the rudder
Failure of components	A physical failure of one or several components leads to failure of the action. The basic events are collected below an or-gate but may be connected through other logic gates. Examples are the failure of a thruster, failure of an engine, failure in the gear box, etc.

TABLE 6: OPERATORS' BASIC FAILURE EVENTS LEADING TO THE FAILURE IN INFORMATION GATHERING AND PRE-PROCESSING

Basic Failure Event	Description
Failure in recognizing data as incorrect	The operator receives incorrect data and fails to recognize it.
Information Miscommunicated	During communication between the operator at the SCC and team member or a third party there may be a miscommunication, in which the information is not complete or in incorrect, or it is sent to the wrong person or at a wrong time.
Data not checked with appropriate frequency	This event is particularly relevant during monitoring tasks. For instance, if the system operates with a high LoA and the operator should take over control in case of a problem, the operator must be checking the HMI

Basic Failure Event	Description
	with an appropriate frequency. If s/he fails to do so, s/he may miss an important shift in one variable, or a variable that is out of the expected range.
Data not obtained (intentional)	The operator intentionally fails to collect a data needed for the operation. S/he may believe, for instance, that the data at hand about the environmental conditions suffices and decides to not collect an additional piece of data that would complete their assessment.
Data discounted	The operator gathers particular data s/he needs but decides to discard it afterwards. S/he may assume the data is not relevant for the situation. For example, s/he may see at their screens that there is an abject approaching the ship, but believe the paths will not cross, and discard this information when performing situation assessment.
Key alert not responded to	A key alert should alert the operator about a crucial status of the system, and their response to it should put them in the path of a successful outcome. For instance, it is expected that in certain LoAs the operator will be able to override the system, or shut it down in case of an emergency, among other situations.
Wrong data source attended to	The operator is aware of a needed information but collect it from a wrong source. This failure event can be particularly relevant in case the operator is monitoring more than one ship at a time.
Reading error	The operator performs an error during reading a piece of information. This may be an information from the HMI or from a written guideline / procedure. They may, for instance, incorrectly read a speed number.
Data misunderstood	The operator gathers data but incorrectly internally processes it.

TABLE 7: OPERATORS' BASIC FAILURE EVENTS LEADING TO SITUATION ASSESSMENT AND DECISION MAKING

Basic Failure Event	Description
Procedure not followed	The operator intentionally does not follow the procedures or guidelines. S/he decide to follow their own knowledge instead, whereas following the procedure / guidelines would put lead to success.
Procedure misinterpreted	The operator is following the procedure or guidelines but incorrectly interprets it.
Procedure step omitted	The operator is following the procedure but omits one step of it. This may be due to, for instance, a perceived lack to available time to follow the procedure, or a confidence that certain steps are not necessary.
Inappropriate strategy chosen	The operator correctly diagnoses the situation but choses an inappropriate strategy to deal with it. For instance, s/he may recognize a potential collision scenario involving the autonomous ship but decide to avoid the collision by lowering the speed when the correct strategy would be to change the ship course.
Decision to delay action	The operator decides to delay an action. This may be because s/he believes that the information at hand is not sufficient and s/he waits for gathering more information.
Inappropriate transfer to a different procedure	The operator transfers to another guideline when it is inappropriate. For example, s/he transfers to a local rule that is not appropriate for the situation in hand.

System state / situation misdiagnosed	The operator misdiagnoses the situation in hand. For instance, s/he may visualize a ship approaching the autonomous ship, but assess that, given its speed and direction, it will not be on collision course.
Failure to adapt procedure to the situation	The operator is following a certain procedure but does not understand how to adapt it to the situation at hand.

TABLE 8: OPERATORS' BASIC FAILURE EVENTS LEADING TO FAILURE IN ACTION

Basic Failure Event	**Description**
Action on wrong component:	The operator performs a correct and needed action, but on the wrong component. The component may be a ship component or a component of the HMI.
Incorrect timing	The operator executes the decision in a bad timing – too late or too early.
Incorrect operation of component	The operator operates the correct component in an incorrect manner.
No action	The operator fails to take the action, despite having previously decided to take it. This may due to external factors.