Fault Tree Analysis of Satellite Constellations

Mareike Metzler Thomas Noll
Software Modeling and Verification Group
RWTH Aachen University, Germany
mareike.metzler@rwth-aachen.de noll@cs.rwth-aachen.de

1 Introduction

The work presented in this paper is part of the joint European-Argentinian research project MISSION[1], which aims to develop advanced technologies necessary to assist in critical aspects of spacecraft design. One of its main goals is to improve reliability and effectiveness for future space missions based on satellite constellations, which play an increasing role. For example, they are intended to be used for monitoring Argentina's maritime areas for illegal activities such as unauthorised fishing [6]. Previous research has examined aspects such as coverage of the observed area, communication dynamics, data handling strategies, and the reliability of individual satellite systems. However, to the best of our knowledge, research addressing the fault behaviour of multiple satellites working together in constellations is scarce. Our work aims to fill this gap by modelling and analysing an exemplary (small) satellite constellation using the AADL and SysML specification languages and the accompanying software tools. More concretely, we assess the capabilities of the COMPASS[2] and SAFEST[3] frameworks with regard to the automated generation of fault trees from systems models and their subsequent evaluation and analysis.

Related work Safety-critical systems, in which failures can be disastrous, are becoming increasingly widespread and complex. Therefore, automated techniques for assessing their safety, dependability, and reliability are becoming increasingly important [3]. One such technique is *fault tree analysis*, which aims to systematically explore all error configurations of a system that eventually lead to its overall failure [10]. As manually creating fault trees is not only inefficient but also prone to human errors, techniques for automatically generating fault trees from system descriptions have been developed [2]. Amongst others, such methods have been applied to the model-based fault analysis of aerospace systems specified in AADL[4] [1, 4] and SysML[5] [8, 9, 11]. However, only single-satellite systems have been considered in this work. To the best of our knowledge, the only formal approach to model-based fault analysis of satellite constellations is described in [5] and employs a stochastic Petri net model.

2 Modelling and Analysis

The satellite constellation case study considered in our work investigates a system composed of three Low-Earth Orbit (LEO) satellites and two ground stations. Each satellite is equipped with an X-band

[1] https://mission-project.eu
[2] https://compass.fbk.eu
[3] https://www.safest.dgbtek.com
[4] https://www.sei.cmu.edu/projects/architecture-analysis-and-design-language-aadl/
[5] https://www.omg.org/sysml/

Presented at MISSION@ETAPS Workshop 2026.

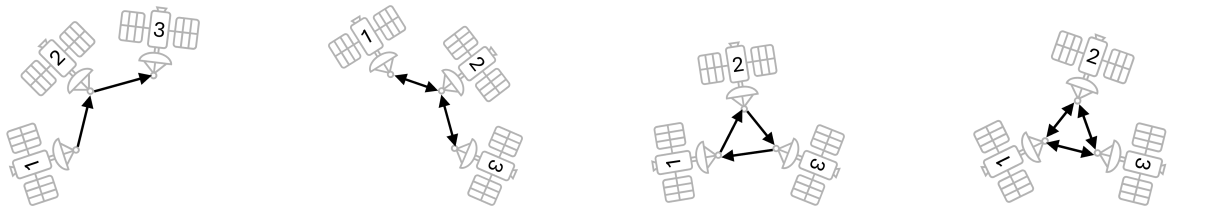
Synthetic Aperture Radar (SAR) payload designed for maritime target detection. The satellites are placed in Sun-synchronous orbits, which ensures constant geometric observation conditions. They can assume different roles: The *leader* (or master) dictates the flight formation and the *followers* (or slaves) adjust accordingly. In the following, we give a brief overview of our model. More details can be found in [7].

System components The satellites are modelled by four components each: The *Ground-to-Space Link (GSL)* enables communication between the satellites and the ground stations. The satellites' capability to communicate with each other is modelled by the *Inter-Satellite Link (ISL)*, which enables the sharing of memory resources, processing functions, and downlink capabilities between the spacecrafts in orbit. The *Data Acquisition Sensor (Data Sensor/Data)* enables the collection and transmission of sensor data to the ground stations. Various other components such as software, power, and propulsion systems are summarised as *Core*. Each of the three satellites possesses all of these components, which are susceptible to failures. If a fault occurs within any of the supplementary systems categorised as Core, the entire satellite becomes inoperable. On the other hand, a failure in either the ISL, GSL, or Data Sensor does not immediately lead to total satellite failure; instead, it may initially require a reconfiguration of the entire system. The ground segment comprises two redundant *Ground Stations (GS)*, with at least one required to maintain system functionality. The failure of a ground station affects all satellites concurrently, as connections to the failed station are lost for all satellites simultaneously.

Operational configuration To ensure our system is operational, at least one satellite must be controllable and capable of collecting sensor data. *Controllable* means that the satellite has a connection to the ground, i.e., that its GSL and at least one ground station are available. Here it is worth mentioning that our analysis models the GSL and ground stations as binary components without explicitly considering the orbital geometry or the contact windows inherent to a non-geostationary LEO mission. Implicitly, continuous availability of the ground segment is assumed, limited only by technical failures rather than operational unavailability due to orbital constraints causing the temporary absence of communication windows. *Collecting sensor data* can either be done independently by the satellite with a ground connection or via another satellite. Each satellite can be either a (unique) *leader* or a *follower*. The leader is responsible for maintaining a connection with the ground to receive commands, which it then can forward to the other satellites. Therefore, the leader needs to have a functional GSL. A follower is capable of acquiring sensor data and can independently downlink this data. Here, we are assuming a scenario in which all satellites are capable of downlinking data to one of the ground stations. If required, the satellites can also change their roles.

Communication topologies The communication between the three satellites can be defined in multiple ways. We have defined four distinct communication topologies, which are visualised in Figure 1. In the *directed chain*, satellite 1 is capable of forwarding commands to satellite 2, which, in turn, can relay commands to satellite 3. However, communication does not occur in the reverse direction, which is additionally possible in the *undirected chain* model. For the *directed ring* topology, the satellites are arranged in a circular configuration, where satellite 1 transmits commands to satellite 2, satellite 2 to satellite 3, and satellite 3 subsequently returns commands to satellite 1. The *undirected ring* again permits communication in both directions among the satellites.

Failure behaviour The most straightforward operational configuration involves the leader receiving commands, acquiring data independently of the followers and downlinking them. If the leader encounters an issue with its Data Sensor, the acquisition command can be passed to the next satellite to collect



(a) Directed chain (b) Undirected chain (c) Directed ring (d) Undirected ring

Figure 1: Communication topologies

the required sensor data. This arrangement is called *one-hop situation*. If the second satellite also experiences a failure in its Data Sensor, the command can then be forwarded to the third satellite, which will collect the data and send it to the ground. This scenario is referred to as a *two-hop situation*. In the event of a failure in the GSL of the leader, the satellite will be unable to fulfil its function, as it will be incapable of receiving commands from the ground. In such a scenario, the function of the leader will be taken over by another satellite equipped with a functional GSL. This process is known as *dynamic reconfiguration*. As the concrete master election mechanism (in particular, the order in which slaves are chosen to become a master) is irrelevant for analysing the general failure behaviour, we refrain from modelling it explicitly.

2.1 SysML and SAFEST

Based on the assumptions established in the previous paragraphs, the satellite constellation system is modelled in SysML. The main task is to manually provide the corresponding fault tree (FT) as annotations to the specification, from which the SAFEST Tool generates a graphical representation. The full SysML specification and more details can be found in [7].

Modelling data acquisition and ground The first step is to establish a top-level event for the FT. In our example, it should be triggered if the overall system is in a Non-Operational Configuration (NOC). Thus, NOC becomes the root node of the overall FT (see Figure 2). The system is modelled from the perspective of the satellites. It is operational if at least one satellite is able to act as a leader, given the necessity of at least one satellite being controllable. This is represented by an `AND` gate at the root of the FT, with child nodes indicating various operational configurations, such as `Sat2 Leader` (see Figure 2). Each satellite qualifies as a leader if it maintains a connection with at least one ground station and can either acquire data independently or via another satellite. Consequently, the nodes (`Sat1 Leader`, `Sat2 Leader` and `Sat3 Leader`) are represented as `OR` gates connecting the respective sub-trees.

To ensure the leader connection to the ground, both a functioning GSL and at least one functional ground station are required. Figure 2 depicts this part of the model for satellite 2. The leader satellite has the capability to acquire data either independently or via another satellite. Initially, a functional Data Sensor of the leader is the only required component. In order to acquire the data via an additional satellite (a follower), it is necessary for both satellites to have functional ISL and for the follower to have a functional Data Sensor.

Adding the Core component Figure 2 illustrates the partial FT for satellite 2 as the leader, as described in the previous paragraph. We see that only three components of satellite 2 are represented in Figure 2: GSL, ISL, and Data Sensor. The other components, summarised as Core, are not included. To account

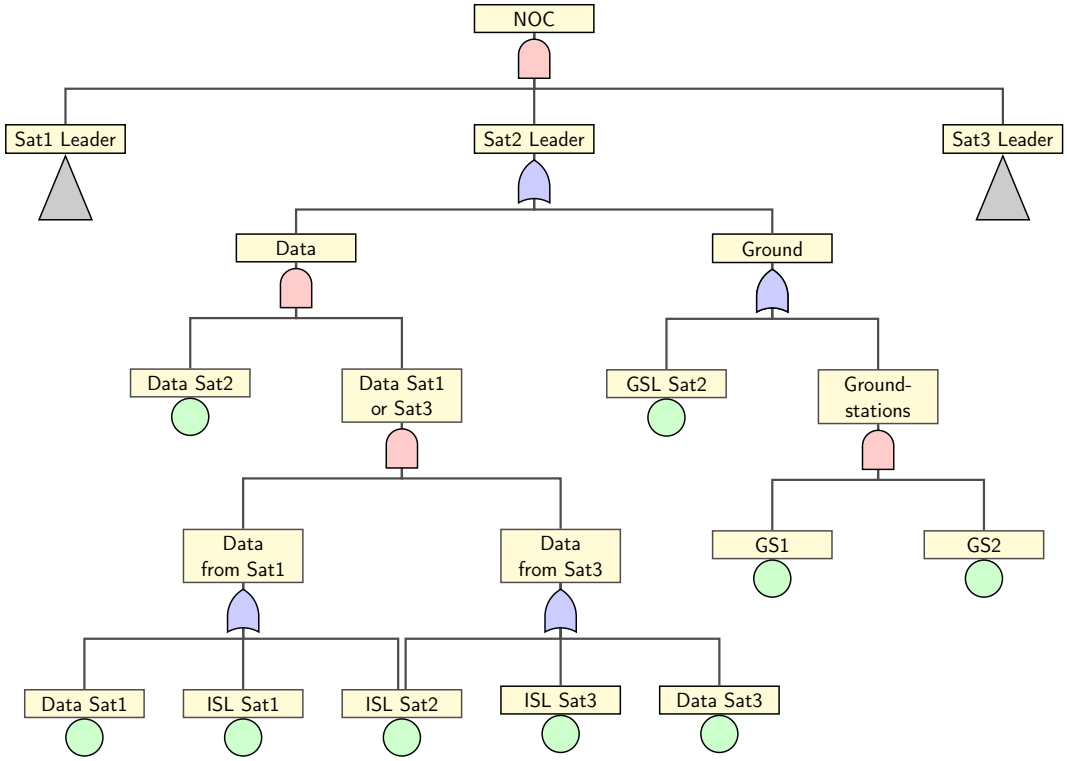


Figure 2: Fault tree for satellite 2 as leader (without Core)

for this, we introduce the basic event `Core Sat2` directly as a child of the node `Sat2 Leader` (see Figure 3). This adjustment is necessary because a failure in any of these components can lead to a total failure of the satellite, rendering it unable to operate as a leader. Moreover, it is important to note that the Data Sensor, GSL, and ISL of the satellite do not automatically fail when `Core Sat2` fails in the FT because the node is just added at the root. However, this is important as those events are referred to by the other satellite models. To maintain a static FT, which is necessary for our analysis, we will use `OR` gates instead of a functional dependency gate (FDEP). Therefore, each of the three satellite components is replaced with an `OR` gate which has the original satellite component and the Core component as a child. Figure 3 illustrates the final partial FT for satellite 2 in its role as a leader.

Implementation As explained before, the SysML specification must be annotated with safety information. The SAFEST Tool provides two dedicated packages for this purpose. The first one, DFTGates, provides all types of gates that can be used to construct fault trees. The second, DFTBEs, defines all basic events that may be used in fault trees and requires the declaration of a failure distribution for each basic event. Since no specific failure rates are known in our application, we assigned an exponential distribution.

2.2 AADL and COMPASS

The alternative approach uses the COMPASS Toolset to analyse the system, which first requires establishing an AADL model using the SLIM language extension. The initial step is to create a generic satellite composed of three parts: a Data Sensor, a GSL and an ISL. The model is illustrated using di-

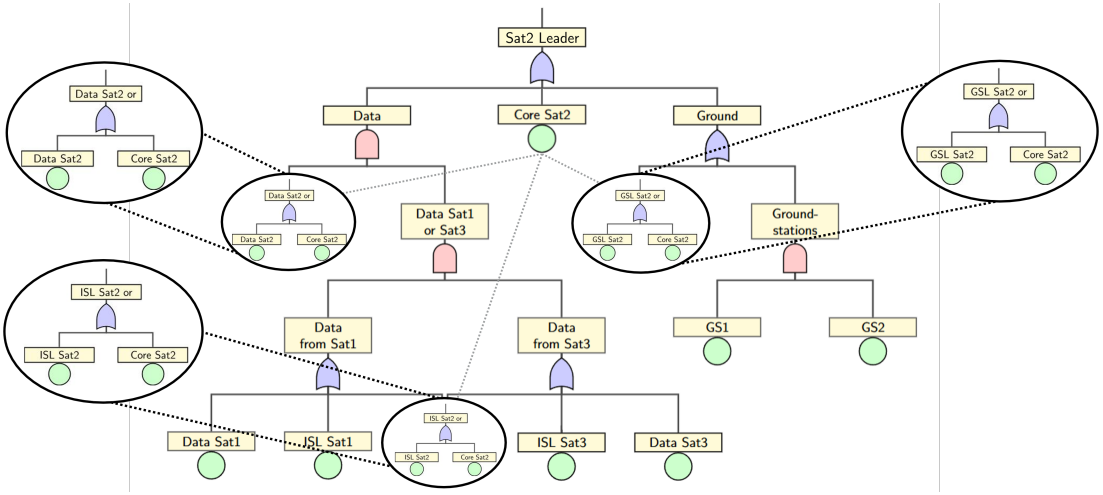
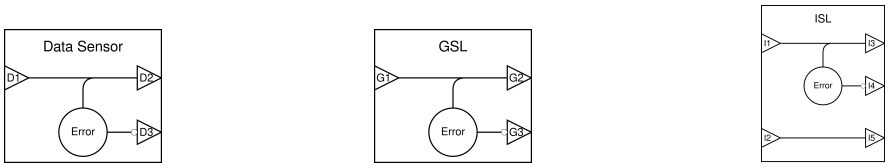


Figure 3: Complete fault tree for satellite 2 as leader

agrams where ports are depicted as triangles. The entire system is modelled using boolean data ports, with the colour of each port indicating its state: green for true and red for false. The lines connecting the ports represent their interconnections, and the triangles should be interpreted as arrows to indicate the direction of these connections. The satellite components with their port numbers are shown in Figure 4. To make the diagrams cleaner and easier to read, the port numbers have been removed later on. Further details can be found in [7].

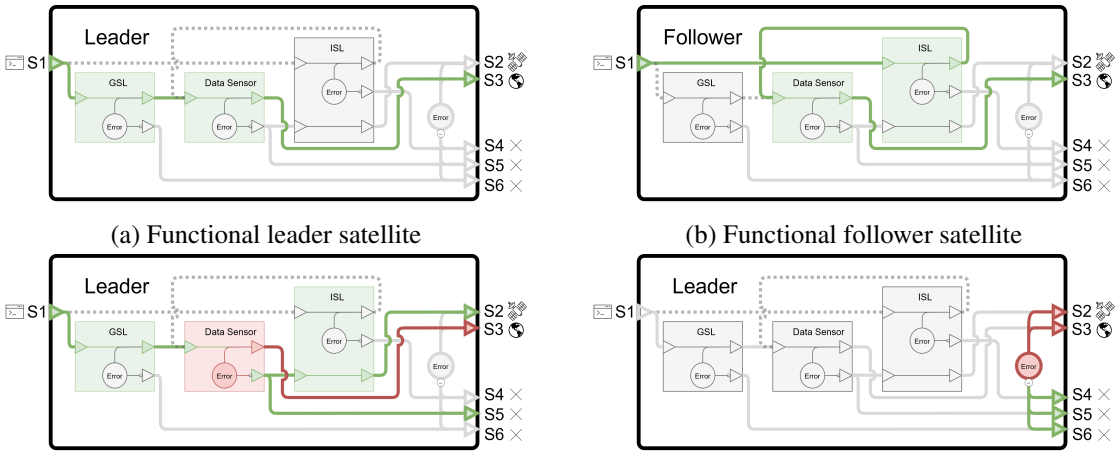


(a) Data Sensor (b) GSL (c) ISL

Figure 4: Satellite components with port numbers

Functional satellites Figure 5a shows an example of a functional satellite acting as a leader. The S1 port is true. This indicates that the command to collect data has been sent. Because the satellite is the leader, the command originates directly from one of the ground stations. This is the reason why it is forwarded to the GSL component of the satellite. Subsequently, the GSL transmits the command to the Data Sensor, which acquires the data and downlinks it to the ground. The status of the sending operation is indicated by the outgoing port S3 coloured in green (true), meaning it is running without failure. The situation is analogous for a follower, as illustrated in Figure 5b. The only difference is that the ISL is used instead of the GSL, because the command originates from another satellite rather than the ground.

Ground segment The ground comprises two redundant stations (refer to Figure 6). Port Gr1 is responsible for sending commands to the satellites, while Gr2 is designated for receiving data. Port Gr3 indicates the overall operational state of the system. This is why Gr3 is employed as the top-level event for the fault tree analysis of the system using the COMPASS Toolset.



(a) Functional leader satellite

(b) Functional follower satellite

(c) Leader with defect Data Sensor

(d) Failure in Core of leader

Figure 5: Satellites models

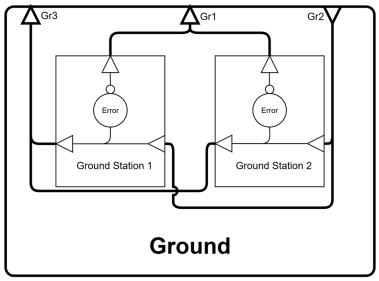


Figure 6: Ground with two ground stations

Satellites with defective Data Sensor We will now examine cases in which the satellites have defective components. Each satellite sub-component has a specific outgoing data port (G3, D3, I4) which indicates a failure when activated. These ports connect to the satellite's outgoing ports (S4–S6), enabling the system to signal which component is malfunctioning and initiate necessary reconfigurations. When a defect occurs in the satellite's Data Sensor, the D2 port of the Data Sensor becomes false, affecting also the S3 port of the satellite (see Figure 5c). As a result, data transmission from this particular satellite to the ground is no longer possible. Meanwhile, port D3 of the Data Sensor becomes true, which is connected to the I2 port of the ISL. The ISL then connects to the S2 port and enables forwarding the command to acquire data to the next satellite. Additionally, the S5 port of the satellite indicates that there is a defect in the Data Sensor. A defective Data Sensor of a follower satellite is handled similarly.

Defective Core If a fault occurs in any of the supplementary systems categorised as Core, the entire satellite becomes inoperable. This is modelled by connecting every outgoing port to the Core component. When a failure occurs, ports S2 and S3 turn false, and the ports S4–S6 turn true to indicate that none of the satellite's components can be used anymore. This situation is illustrated in Figure 5d. A similar behaviour occurs with a follower satellite.

Satellite connections Next, we connect the satellites according to the undirected ring communication topology. This means that the S2 port of satellites 1 and 3 is connected to the S1 port of satellite 2, and that the S2 port of satellite 2 is connected to the S1 port of satellites 1 and 3.

Implementation The SLIM language extension of AADL includes mode and state transitions to facilitate the implementation of the different operational modes used by satellites and their components. Unfortunately, during simulation in the COMPASS Toolset, significant delays occurred in the mode transitions of the satellites and their components. This made analysing the model and developing a functioning version quite challenging. Hence, we modelled the modes as ports that are activated when the instance is in that specific mode. The mode changes are implemented by a controller in our model, which connects all these ports. Once the modelling process is complete, COMPASS is used to generate a fault tree. A part of this generated FT is shown in Figure 7.

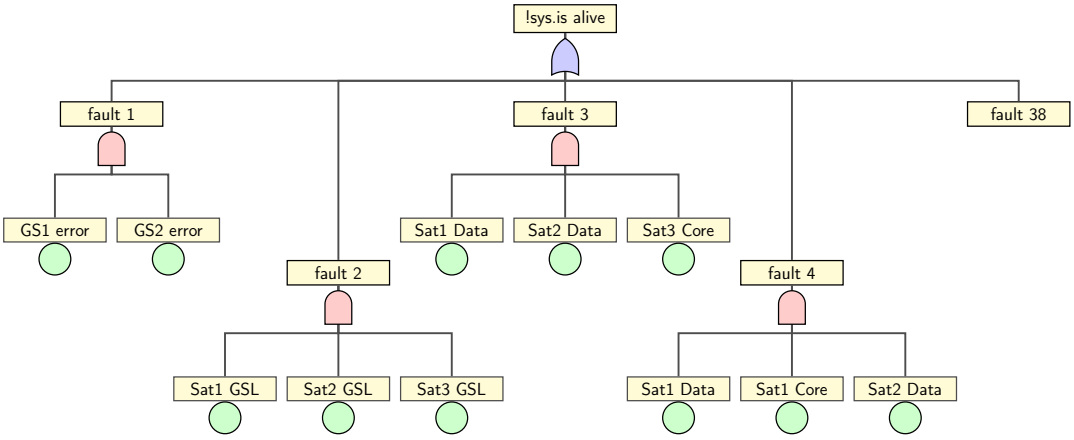


Figure 7: Part of the fault tree generated by COMPASS

3 Evaluation

As seen before, after modelling the satellite constellation system in AADL and SysML, the resulting fault trees can be generated by COMPASS and SAFEST. Their visual representation shows notable differences. The FT extracted by the SAFEST Tool from manual annotations to the SysML code exhibits a hierarchical structure, whereas the model-checking algorithm employed by COMPASS yields a flatter and more horizontally-oriented FT. Also, the number of nodes varies significantly between both. The FT generated by COMPASS comprises 166 nodes, while the SAFEST Tool only produces 44 nodes. This difference arises from COMPASS's approach to directly calculating the minimal cut sets (MCS) and presenting them as an FT: Each level-one node, labelled `fault1`, `fault2`, `fault3`, etc. in Figure 7, corresponds to an MCS of the system. MCS can also be computed by the SAFEST Tool by applying standard algorithms to the FT extracted from the specification. We note already here that the MCS calculated by SAFEST are identical to those derived in COMPASS using model checking, i.e., both tools yield consistent results. In the following, we provide an in-depth analysis of the MCS.

Basic analysis The first minimal cut involves the failure of both ground stations and is the smallest cut. All other minimal cuts are related to satellite components. To give a systematic overview, they are

Satellite 1	Satellite 2	Satellite 3
Core	Core	Core
Data	Data	Data
GSL	GSL	GSL
Core	GSL	GSL
GSL	Core	GSL
GSL	GSL	Core
GSL	Core	Core
Core	GSL	Core
Core	Core	GSL
Core	Data	Data
Data	Core	Data
Data	Data	Core
Data	Core	Core
Core	Data	Core
Core	Core	Data

(a) Standard cuts

Satellite 1	Satellite 2	Satellite 3
GSL ISL	Data	Data
Data	GSL ISL	Data
Data	Data	GSL ISL
ISL Data	GSL	GSL
GSL	ISL Data	GSL
GSL	GSL	ISL Data

(b) Further typology-invariant cuts

Table 1: Minimal cut sets for satellite component failures

organised as shown in Table 1 with one column for each satellite and each row representing a distinct minimal cut. The table is organised into groups where the minimal cuts within a group differ only by permutation. We designate one representative for each group, which is highlighted in grey. In the analysis, 15 instances were identified where all satellites are either malfunctioning, unable to collect data, or incapable of connecting with the ground. In such scenarios, achieving an operational configuration is not feasible any more.

In Table 1b, six additional scenarios are listed in which the system cannot operate regardless of the communication topology used. In the first three cases, one satellite is still capable of acquiring data but is unable to connect to the ground (due to a GSL failure) or to receive commands from the other satellites (due to an ISL failure). In the second group, only one satellite can establish a ground connection (with its functional GSL), but it cannot forward commands to another satellite because of an ISL failure.

Taking communication topology into account To demonstrate the influence of the communication topology on the failure behaviour, we compare the minimal cuts for different types of communication, beginning with the directed and undirected ring (see Section 2 for a description of communication topologies). Table 2 illustrates the resulting differences. In the case of the undirected ring, an additional failure in the ISL is necessary for the entire system to fail (see Table 2a). This is because communication can be forwarded in both directions in this topology.

Even in the cases shown in Table 2b, the failure of an ISL component is necessary for an overall system failure. But now there are two potential satellites where an ISL must fail, so there are two different scenarios. They are represented by the same background colour. This clarifies the reason behind the observed difference in the number of minimal cuts: 37 in the case of undirected ring versus 40 for directed ring.

Next, we will examine minimal cuts of the undirected chain, which was chosen as the communication topology of Section 2.1. Beyond the scenarios illustrated in Table 1, there are additional 16 minimal cuts for the undirected chain presented, as shown in Table 3. Specifically, ten of these cases arise due to the failure of the ISL component in the middle satellite (see Table 3a). Unfortunately, the other two satellites are not fully operational on their own, making it necessary to forward commands through satellite 2. However, since this is not feasible, the configuration is rendered non-operational. The first six cases are

Directed ring			Undirected ring		
Satellite 1	Satellite 2	Satellite 3	Satellite 1	Satellite 2	Satellite 3
GSL ISL	GSL	Data	GSL ISL	GSL ISL	Data
Data	GSL ISL	GSL	Data	GSL ISL	GSL ISL
GSL	Data	GSL ISL	GSL ISL	Data	GSL ISL
ISL Data	GSL	Data	ISL Data	GSL	ISL Data
Data	ISL Data	GSL	ISL Data	ISL Data	GSL
GSL	Data	ISL Data	GSL	ISL Data	ISL Data

(a) Additional ISL failure in undirected ring

Directed ring			Undirected ring		
Satellite 1	Satellite 2	Satellite 3	Satellite 1	Satellite 2	Satellite 3
GSL	Data	Core			
Core	GSL	Data			
Data	Core	GSL			
GSL ISL	Core	Data	GSL ISL	Core	Data
			GSL ISL	Data	Core
Data	GSL ISL	Core	Data	GSL ISL	Core
			Core	GSL ISL	Data
Core	Data	GSL ISL	Core	Data	GSL ISL
			Data	Core	GSL ISL
ISL Data	GSL	Core	ISL Data	GSL	Core
			ISL Data	Core	GSL
Core	ISL Data	GSL	Core	ISL Data	GSL
			GSL	ISL Data	Core
GSL	Core	ISL Data	GSL	Core	ISL Data
			Core	GSL	ISL Data

(b) Further different minimal cuts

Table 2: Comparison of minimal cuts from directed and undirected ring

solely related to the types of communication employed. The remaining four cases can also be identified for the undirected ring. The same applies to the six additional cases shown in Table 3b.

Criticality analysis A common analysis question is how critical the failure of a particular component is for the overall system reliability. As we do not have quantitative information about the failure rates of specific components, we follow a "qualitative" approach based on MCS to address this issue. More exactly, we have created a statistics about the frequency of each component of every satellite occurring in the minimal cuts. The results are shown in Table 4 for the undirected chain topology. The data indicates that -– not surprisingly -– satellite 2 plays a crucial role in the undirected chain communication topology, along with the GSL and Data Sensor component for the satellites in general.

4 Conclusion

Our results show that qualitative analyses of the fault trees generated from manually designed system specifications are feasible and yield interesting results. However, additional information such as probabilistic failure rates of a system's basic components is required in order to enable a full quantitative analysis of reliability metrics such as Mean Time To Failure (MTTF), and to derive more precise conclusions about the satellite system design. As a starting point, (approximate) rate values obtained from literature or space reliability databases could be used to compare architectural alternatives in a more informed manner.

Undirected chain		
Satellite 1	Satellite 2	Satellite 3
GSL	Core	Data
Data	Core	GSL
GSL	ISL GSL	Data
Data	ISL GSL	GSL
GSL	ISL Data	Data
Data	ISL Data	GSL
Data	ISL GSL	Core
Core	ISL GSL	Data
GSL	ISL Data	Core
Core	ISL Data	GSL

(a) Minimal cuts for defect ISL of satellite 2

Undirected chain		
Satellite 1	Satellite 2	Satellite 3
GSL ISL	Data	Core
Core	Data	GSL ISL
ISL Data	GSL	Core
Core	GSL	ISL Data
GSL ISL	Data	GSL ISL
ISL Data	GSL	ISL Data

(b) Further cases

Table 3: Additional minimal cuts for undirected chain

	Satellite 1	**Satellite 2**	**Satellite 3**
GSL	13	14	13
ISL	6	10	6
Data	13	14	13
Core	11	9	11
Sum	43	47	43

Table 4: Frequency of components in minimal cut sets for undirected chain

The frameworks COMPASS and SAFEST employed in this case study exhibit different strengths and weaknesses. The former is a dated tool, but it allows automatic fault tree generation. The representation of the FT is less intuitive in structure, but the minimal cut sets can be read directly from the tree. The operation of the tool is often cumbersome, due to outdated software and the implementation as a virtual machine. SAFEST is more modern, but it does not (yet) offer a fully automatic generation of FTs from the system specification. Rather, it extracts a graphical representation of FTs from user-given annotations in the SysML specification.

Acknowledgment Our work has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 101008233.

References

[1] Joshi Anjali, Steve Vestal & Pam Binns (2007): *Automatic generation of static fault trees from AADL models*. In: *DSN 2007 Workshop on Architecting Dependable Systems*. Available at `https://hdl.handle.net/11299/217313`.

[2] Axel Berres & Holger Schumann (2016): *Automatic generation of fault trees: A survey on methods and approaches*. In: *Risk, Reliability and Safety: Innovating Theory and Practice*, chapter Systems Reliability Models and Applications, CRC Press, doi:10.1201/9781315374987.

[3] Marco Bozzano & Adolfo Villafiorita (2010): *Design and Safety Assessment of Critical Systems*. CRC Press, doi:10.1201/b10094.

[4] Marie-Aude Esteve, Joost-Pieter Katoen, Viet Yen Nguyen, Bart Postma & Yuri Yushtein (2012): *Formal Correctness, Safety, Dependability and Performance Analysis of a Satellite*. In: *Proc. ICSE*, ACM and IEEE, pp. 1022–1031, doi:10.1109/ICSE.2012.6227118.

[5] Daniel Farias, Bruno Nogueira, Ivaldir Farias Júnior & Ermeson Andrade (2025): *A modeling-based approach for dependability analysis of a constellation of satellites*. Software and Systems Modeling 24, pp. 209–224, doi:10.1007/s10270-024-01197-7.

[6] Juan A. Fraire, Santiago Henn, Gregory Stock, Robin Ohs, Holger Hermanns, Felix Walter, Lynn Van Broock, Gabriel Ruffini, Federico Machado, Pablo Serratti & Jose Relloso (2024): *Quantitative analysis of segmented satellite network architectures: A maritime surveillance case study*. Computer Networks 255, p. 110874, doi:10.1016/j.comnet.2024.110874.

[7] Mareike Metzler (2025): *Evaluating Tool Support for Fault Tree Analysis of Satellite Constellations*. Bachelor's thesis, RWTH Aachen University, Aachen, Germany, doi:10.18154/RWTH-2025-09682.

[8] Faïda Mhenni, Nga Nguyen & Jean-Yves Choley (2014): *Automatic fault tree generation from SysML system models*. In: 2014 IEEE/ASME International Conference on Advanced Intelligent Mechatronics, IEEE, pp. 715–720, doi:10.1109/AIM.2014.6878163.

[9] Faïda Mhenni, Nga Nguyen & Jean-Yves Choley (2018): *SafeSysE: A Safety Analysis Integration in Systems Engineering Approach*. IEEE Systems Journal 12(1), pp. 161–172, doi:10.1109/JSYST.2016.2547460.

[10] Enno Ruijters & Mariëlle Stoelinga (2015): *Fault tree analysis: A survey of the state-of-the-art in modeling, analysis and tools*. Computer Science Review 15–16, pp. 29–62, doi:10.1016/j.cosrev.2015.03.001.

[11] Zhihong Zhao, Renfei Li & Tao Meng (2025): *A MBSE-based fault modeling approach for satellite*. Journal of Physics: Conference Series 2977(1), p. 012077, doi:10.1088/1742-6596/2977/1/012077.