

bradscholars

Adaptation of Model Transformation for Safety Analysis of IoT-based Applications

Item Type	Conference paper
Authors	Abdulhamid, Alhassan;Kabir, Sohag;Ghafir, Ibrahim;Lei, Ci
Citation	Abdulhamid A, Kabir S, Ghafir I and Lei C (2023) Adaptation of Model Transformation for Safety Analysis of IoT-based Applications. The UNified Conference of DAMAS, IncoME and TEPEN Conferences. August 2023. University of Huddersfield. Mechanisms and Machine Science, vol 152.
DOI	https://doi.org/10.1007/978-3-031-49421-5_79
Rights	© 2023 Springer. This version of the article has been accepted for publication, after peer review (when applicable) and is subject to Springer Nature's AM terms of use. The Version of Record is available online via the doi below
Download date	2026-03-06 00:02:30
Link to Item	http://hdl.handle.net/10454/19577

Adaptation of Model Transformation for Safety Analysis of IoT-based Applications

Alhassan Abdulhamid, Sohag Kabir, Ibrahim Ghafir, and Ci Lei

Department of Computer Science, University of Bradford, Bradford, UK
a.abdulhamid2; s.kabir2; i.ghafir; c.lei1 {@bradford.ac.uk}

Abstract. The Internet of Things (IoT) paradigm has continued to provide valuable services across various domains. However, guaranteeing the safety assurance of the IoT system is increasingly becoming a concern. While the growing complexity of IoT design has brought additional safety requirements, developing safe systems remains a critical design objective. In earlier studies, a limited number of approaches have been proposed to evaluate the safety requirements of IoT systems through the generation of static safety artefacts based on manual processes. This paper proposes a model-based approach to the safety analysis of the IoT system. The proposed framework explores the expressiveness of UML/SysML graphical modelling languages to develop a dynamic fault tree (DFT) as an analysis artefact of the IoT system. The framework was validated using a hypothetical IoT-enabled Smart Fire Detection and Prevention System (SFDS). The novel framework can capture dynamic failure behaviour, often ignored in most model-based approaches. This effort complements the inherent limitations of existing manual static failure analysis of the IoT systems and, consequently, facilitates a viable safety analysis that increases public assurance in the IoT systems.

Keywords: Internet of Things, Smart Home, Safety Assurance, Failure Analysis, Model-Based System Engineering, Dynamic Fault Tree

1 Introduction

The Internet of Things (IoT) is one of the research topics that has received much attention from industry and academia over the last decade due to its ability to provide valuable services across different domains [1]. The rapid growth of IoT applications has continued to revolutionise the world with innovative and intelligent solutions providing convenience and enhanced efficiency [2,3]. As the efficiency of the IoT ecosystem continues to prosper, some emerging challenges need to be overcome to have dependable systems. For instance, IoT systems are increasingly evolving, becoming more dynamic in behaviours, adaptive, cooperative and autonomous [4]. The greater degree of autonomy of the IoT systems in decision-making power to perform critical tasks with minimal human intervention inevitably comes with increasing safety concerns. Additionally, because of the nature of deployment and

many IoT applications operating in harsh and remote locations, IoT sensing and other devices are vulnerable to an increased rate of failure, which can put a system in a precarious state [1]. To address safety concerns and meet the growing progress in IoT design, it is essential that safety analysis frameworks of IoT design also evolve at the same pace to contribute to the viable analysis and verification process.

Safety analysis of IoT systems is an integral design requirement to identify potential safety-related issues and ascertain whether the system is safe to operate [5]. Numerous analysis frameworks were developed to evaluate the possibility of safety violations in the design process through rigorous evaluation of the system's architecture, components, configurations, behaviours, operating states, and conditions [6]. The two most used frameworks in various safety-critical domains, including the IoT, are Fault Tree Analysis (FTA) and Failure Mode Effects Analysis (FMEA) [1]. The FTA framework draws its strengths in its qualitative and quantitative analysis of the overall system failure occurrence using various decompositions of hierarchical events and corresponding logical gate symbols [7,8]. Although FTA and FMEA approaches have been vastly successful in their analysis nature over the decades and across numerous safety-critical domains, they have significant limitations, such as manual nature (documents-based), and their analysis is restricted to static systems failure conditions [9]. These limitations made them more susceptible to cumbersome human errors and lacked support for re-usability, an essential feature of the system engineering process [1].

As IoT design is fast progressing, static and manual analysis approaches must be improved to guarantee IoT innovations' safety. Accordingly, research to explore the expressiveness of model-based system engineering (MBSE) to develop a more viable analysis framework of IoT systems is gradually coming onboard the domain [10]. Although this effort has begun to yield results in other domains, it is still in its infancy in the IoT domain [11]. The motivation of this paper is to improve the static and document-based nature of the FTA framework by exploring the links between system architecture and the analysis models to generate a model-based dynamic framework. This will contribute to the rapid progress of IoT design and increase public confidence in the systems.

The rest of the paper is organised as follows. Section 2 provides relevant background and reviews some of the existing studies. Section 3 provides a detailed description of the proposed approach, and Section 4 demonstrates an illustrative example of the proposed approach using IoT-enabled SFDS. Finally, Section 5 summarises the key aspects and suggests future research directions.

2 Background and Literature Review

As modern systems evolve in sophistication and functionalities, the systems' dynamic behaviours give rise to increasing dynamic failure characteristics, which safety engineers must address to obtain a dependable system. To meet this demand, an extension of the FTA framework known as the Dynamic Fault Tree (DFT) approach was developed to capture dynamic failure behaviours [12]. The approach augments the FTA with dynamic gates such as Priority AND (*PAND*), Priority OR

(*POR*), Functional Dependency (*FDEP*), Sequence Enforcing Gate (*SEQ*) and *SPARE* gates. These new gates provide greater flexibility in the safety analysis of dynamic systems by capturing various time, spare and functional-dependent failure behaviours of systems [9,10]. Despite this remarkable achievement of the DFT approach, the manual analysis process of developing the DFT can be cumbersome, time-consuming, and more prone to human errors, leading to inconsistency and incompleteness in the safety analysis of IoT systems [1,8].

With further progress in system analysis and verification processes, the MBSE approaches emerge based on the formal and tools-based system models to analyse different aspects of IoT systems' functional and NFPs [13,14]. Many of these formal methodologies are created utilising the expressiveness and functionalities of modelling languages through harmonised engineering and domain-specific models. Numerous studies have been performed to analyse various NFPs of systems using MBSE approaches where some safety analysis artefacts were automatically generated. Based on the literature, FTA and FMEA were developed based on the MBSE process, including, but not limited to, [10,13,15-17]. The research of Alshboul and Petriu [13] proposed an integrated safety analysis of electric kettle systems within the MBSE development process. The approach only considered static failure analysis of simple mechatronic systems using Boolean *AND* and *OR* gates. The limitation of static safety analysis using Boolean *AND* and *OR* gates is that it cannot model several complex failure attributes of the IoT system, which are necessary for a viable safety analysis. Although some modest contributions were made within the realm of MBSE to automate

DFT in refs. [10,11], these studies are still proposed in the context of generic systems engineering and industrial control systems. However, the unique nature of the IoT systems having complex systems failure from both physical and cyber components, demands a viable analysis framework that will be model-based and has the flexibility of dynamic and adaptive failure analysis.

Drawing from the stated limitations of the existing analysis frameworks, the proposed approach in this paper seeks to bring a novel idea of conducting model-based static and dynamic failure analysis of IoT-based applications. The proposed approach's advantage lies in its composition, which enables an easy understanding of the effects of each component and internal configuration of the system in ascertaining the overall IoT system's safety assurance. This supports re-usability, reduces human errors, and supports iterative system design.

3 System Design and Safety Analysis

Fig. 1 shows the proposed framework for safety analysis of the IoT system using the MBSE approach. The following subsections provide a high-level description of the framework.

3.1 Source Modelling of the IoT System in the MBSE Environment

In developing the safety analysis model in the MBSE environment, the IoT system under consideration must be represented in the selected modelling environment using appropriate diagrams, links, and profiles. The proposed approach is based on UML/SysML meta-models. The SysML is a lightweight dialect (extension) of UML designed for systems engineering applications. The Flowchart in Fig.1 represents the steps to follow to model the system and generate the DFT for safety analysis of the IoT system. Based on Fig.1, the steps are summarised in the proceeding subsections for brevity.

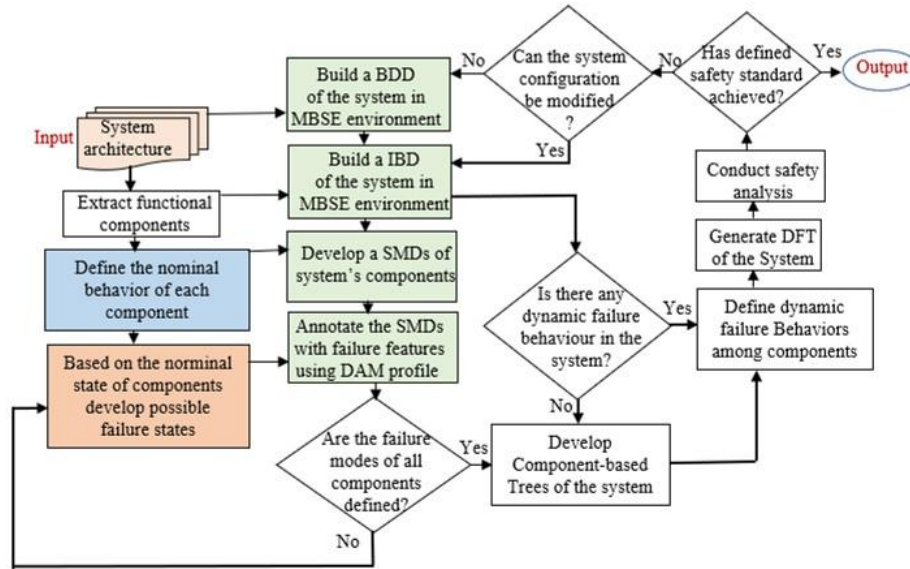


Fig.1. Flowchart of the proposed approach

Understand the System's Safety Requirements IoT systems have different design considerations based on their applications and other features. The first step is to understand the system's safety requirements under consideration. This entails what the system is all about, its components and intended functionalities, among others. The system's proposed or existing design architecture will be studied at this stage. Various components that compose the system are to be extracted for subsequent analysis. A requirement diagram in the SysML environment can be helpful for this purpose but is optional.

Develop a Static Model of the System Once the design architecture of the system is obtained, the system should be decomposed into components, hierarchies, and dependencies. The system's various compositions and hierarchical structure are represented using the SysML environment's Block Definition Diagram (BDD). The system is decomposed into components using predefined blocks which logically

represent the hardware and software of the systems. Also, the relationship in terms of dependencies, generalisations, associations, aggregation, and inheritance is defined between components and the overall design.

Develop the Internal Structure Modelling of the System The system components should be connected using the Internal Block Diagram (IBD) of SysML to show how the elements within the system are connected and the information flows between components. The internal configuration of the system shall show how the various components are connected using ports and data flow symbols.

Develop Nominal and Failure Modelling of the System In this step, the behavioural features modelling of the system entails the representation of the functional(nominal) and failure features of the system. The system behaviour is decomposed based on the behaviours of its constituent components. The failure modelling of the source model generation represents how the system elements could fail using state machine diagrams (SMDs). Each component's nominal and failure states and the corresponding transitions and triggers are developed using component-based SMDs. A lightweight extension of the SMDs meta-model was represented using the DAM profile based on *DaStep* stereotypes [13]. The stereotype and its various tag values represent the system's failure and error states and the components' transitions and triggers.

3.2 Transformation of MBSE Source Model into Safety Analysis Artefact

The transformation process involves logically converting a source model into a target model. UML/SysML diagrams are transformed into a formal safety analysis model using various methodologies. Notably, a pattern-based transformation approach is the simplest in a less complicated system and was proposed in ref. [18]. Other research suggests the automatic generation of target models using various programming languages and tool-based approaches. In these tool-based approaches, the source model of the system is exported as an XMI file to other suitable tools as input for the transformation. In this paper, we adopted a pattern-based transformation approach to generate our DFT from the existing transformation approaches. The transformation process in this paper is conducted in three folds.

Component Level Fault Tree Generation The development of component-based failure analysis artefacts of the IoT system was achieved based on the various component failures annotation conducted using SMDs. This approach supports a painstaking analysis of the contribution of each component to the overall system failure and also supports the iterative design process. A component with high failure probability can be changed or reinforced with an extra redundant element. A component-based FT model is generated based on SMDs annotated with DAM profiles at this stage. Each system component under consideration is transformed to a corresponding component level FT. Various trees of each component are generated at this level for subsequent analysis.

Dynamic Failure Behaviour Mapping This step involves examining the dynamic failure characterises between components considering the system as a whole. For

simplicity, we assume a dynamic failure will only happen between components. However, this may not hold in complex systems, as dynamic failure within a component is possible. We hope to consider this in our future work. Nevertheless, adding dynamic failure consideration at the system level generated valuable information about system safety beyond what is obtained from the static component trees. At this stage, a cross-analysis of the IBD of the system is conducted based on these checklists.

- Map instances of the IBD where concurrent failure propagation occurs from one trigger event, leading to multiple dependent failure events as an *FDEP* gate.
- Map instances of the IBD where failure events occur in a specific order for inevitable undesired intermediate or top-level failures to occur as a *PAND* gate.
- Map instances of the IBD where a particular failure event must occur first but does not require all other events to cause an undesired intermediate or top-level failure as a *POR* gate.
- Map instances of the IBD for sequential failure, leading to an undesired event as an *SEQ* gate.
- Map instances of IBD to model redundancy of component failures as a *SPARE* gate.

Dynamic Fault Tree Generation This is the penultimate stage of our approach, where the overall DFT of a system under consideration is developed. This is done by the composition of component-level trees and adding the dynamic gates defined based on the system's dynamic failure behaviours. At this stage, the overall failure of the system of interest is now the top undesired event of the dynamic tree. In contrast, the previous top-level events in each component FTs become the intermediate event.

3.3 Safety Analysis

The safety analysis framework aims to determine the various ways the system could be faulty or erroneous based on the individual failure of components and failure propagation between components within the system. Safety analysis using DFT is conducted using qualitative and quantitative approaches. The qualitative process involves identifying minimum essential failure events known as Minimum Cut Sets (MCSs), which directly impact the occurrence of the undesired top event. MCSs are obtained using various algorithms, notably the Method of Obtaining Cut Set (MOCUS) [8]. The quantitative evaluation provides the probability of the top event, obtained from the disjoint sum of the likelihood of the MCSs. The DFT generated using this framework can be converted to other models such as Bayesian Network, Petri-net, and Monte Carlo simulation for quantitative safety analysis, as demonstrated in [12]. This will be considered in our future works.

4 Illustrative Example

To demonstrate the efficiency of the proposed dynamic failure analysis framework, we considered a hypothetical case study of the IoT-enabled SFDS.

4.1 System Description

The system is a safety-critical case study that involves the application of the IoT to detect, prevent and combat fire incident hazards in an intelligent home. Fig. 2 shows a high-level overview of the entire system. The system consists of two sets of battery-enabled wireless sensor networks (WSN A and B), and each of the sensors has specific battery and wireless interfaces. Each WSN has a gateway node, which passes sensing data to a dedicated smart hub (SH) and their health status to a smart monitor system (MS). The Hub passes actuation processes to a sprinkler system (SS), alarm system (AS), and intelligent switch of central heating (SS to CHS). It informs the remote user via the internet protocol.

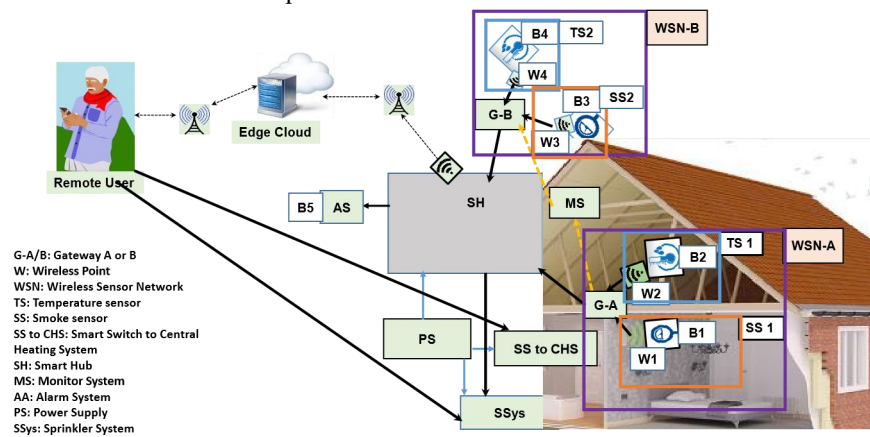


Fig.2. IoT-enabled SFDS

Due to space limitations, this paper will demonstrate the proposed framework's effectiveness using only the WSNs and MS components. The demonstration of the entire system failure behaviour and comprehensive safety analysis involving all the components will be demonstrated in our future work. Accordingly, the expected functional behaviours of the selected parts of the SFDS under consideration are as follows.

- The WSN A is the primary sensing component, which is expected to generate sensing data (temperature and smoke particles) as environmental inputs and send the reading to the SH.
- The WSN B will remain in standby mode and will only be activated by MS in the event of failure of WSN A. Upon activation, WSN B will do the same function as WSN A, and MS will operate vice versa.
- The MS will continuously receive data from the WSNs to ascertain the health status of the sensors. If faults or errors are detected, the redundant sensor network

will be activated to ensure a continuous flow of accurate data to the SH for necessary safety prevention and mitigation actions.

Based on this functional behaviour of the sub-system enumerated, the task of safety engineers is to first determine the failure condition of the sub-system under analysis (WSNs and MS sub-systems). At this level, the failure condition to be analysed is “*the omission of accurate sensing data at the inputs of SH*”. As demonstrated in the subsequent subsections, our approach will be followed to achieve this failure condition.

4.2 Extraction of Functional Components and Safety Considerations

Considering our outline based on Fig. 1, the first step is to understand the safety requirements and design considerations proposed by the system engineers. The two components under consideration are WSN and MS. The safety considerations taken into the design are as follows:

- The sub-system under consideration is only safe if there is uninterrupted data flow from the sensor systems to the CH for safety analysis and actuation of safety processes to prevent and curtail fire incidents in the smart home.
- Each of the two WSNs (A and B) comprises two battery-enabled sensor nodes with wireless interfaces as sub-components.
- The smart Hub receives sensing data directly from the sensor gateways, not from the MS, to guard against single-point failure. The monitor and the Hub are connected to the central power system of the smart home.

4.3 Static Configuration Modelling

This involves the development of a formal composition of the system using the MBSE environment. In our case, we use SysML diagrams to develop the BDD of the IoT system using Papyrus based on Eclipse IDE 2022-23. Although other diagrams, such as class and deployment diagrams in the UML, are also suitable. The BDD of the IoT-enabled SFDS components under consideration is shown in Fig. 3. In the BDD, the overall conceptual system SFDS is the context. The context is linked with WSN A, WSN B and MS using composite aggregation to depict their dependency.

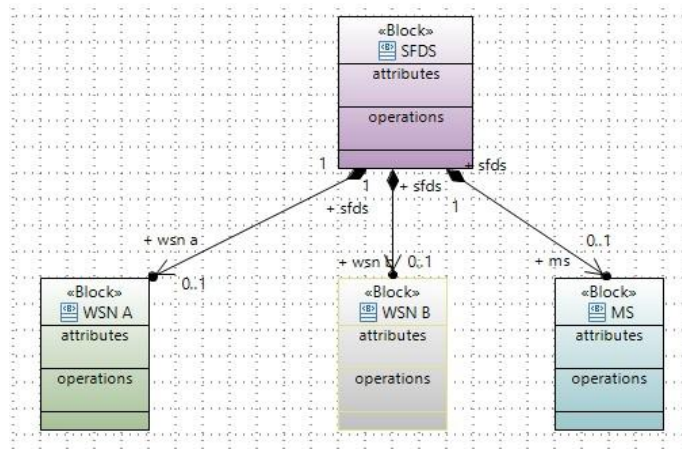


Fig.3. Block Definition Diagram of the Considered Part of the SFDS

4.4 Internal Configuration Modelling

Next is the IBD of the system, which shows the exploded view of the context developed in the BDD. The IBD of the system represents how components cooperate and interact to make the system's functional behaviour. The IBD of the three components represents the data flow direction pointed by the arrows, as shown in Fig. 4. The IBD was created to show WSN A, WSN B and MS interaction. They are all represented within the context blocks of SFDS. The interactions within the system are represented using ports and item flow symbols showing the data flow within the system.

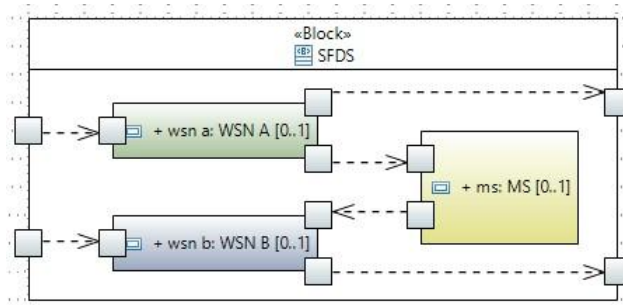


Fig.4. Internal Block Diagram of the Considered Part of the SFDS

4.5 Failure Annotation of the System

At this stage, the functional model of the system was created using SMDs meta-model and extended with a DAM profile in the DICE simulation environment to model the component-based failure behaviour. The functional and failure model of the WSN components is shown in Fig. 5. From the diagram, the sensor component has subcomponents, and their failure events are the basic events leading to the fault or

error reading of the environment. These are the sensing unit, battery unit, wireless components, microcontroller and sensor gateway node. Using the SM features, each component's nominal and failure characteristics are represented as a state. As demonstrated, we represent the WSN based on three nominal states, which are shown in green colour. These are capturing the sensor readings, handling the data collected, and sending the sensor readings to SH. However, the failure states are changes of nominal states based on failure or error transition triggered by some primary conditions. The failure, error states, transitions and triggers are annotated using the *DaStep* stereotype built into a DAM Profile. Similarly, the WSN B and MS were also modelled according to their functional behaviour representations as states, transitions, and triggers.

4.6 Component Tree Generation

The transformation of component-based SMDs to CFT was carried out using a pattern-based approach. In doing so, only the failure states from the developed SMDs and their corresponding transitions and triggers were used. The transformation of a component source model to a fault tree is shown in Fig. 6. From the diagram, the nominal conditions in the SMDs are only helpful in generating the failure states of each component. The overall malfunction of the component is represented as a failure condition under analysis in the corresponding CFT. Each failure or erroneous state of a component is transformed into an event in the CFT connected with appropriate logical gates. The various triggers leading to the different failures or erroneous states are considered as the basic events of the tree.

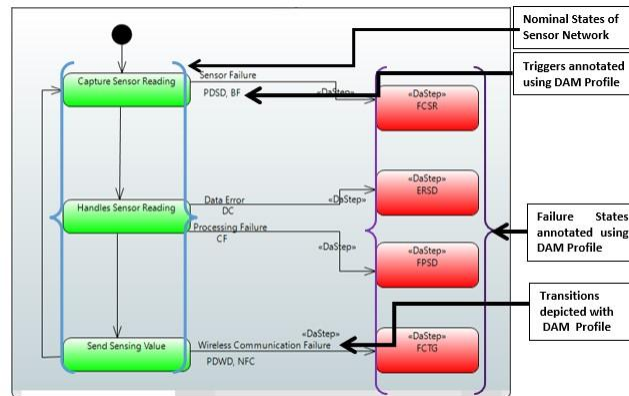


Fig.5. Failure annotation using SMD and DAM Profile

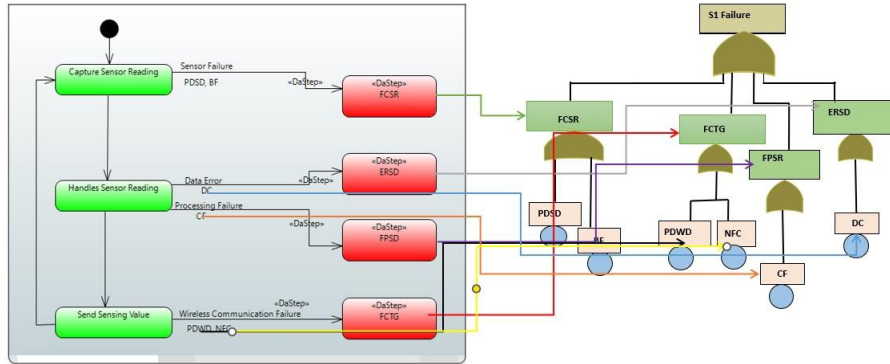


Fig.6. Source Model Transformation Using Failure Mapping Pattern

4.7 Dynamic Failure Behaviour Analysis of the System

After careful analysis of the IBD developed, some of the dynamic features of the internal configuration leading to dynamic failure were established. We established two patterns of failures between the WSNs and the monitor components in the system based on the item flows depicted in the IBD. These failure event combinations are $WSN A \cdot WSN B$ or $MS < WSN A$. The logical explanations are as follows:

- The two WSNs (A and B) can both fail and once this happens, there will be no sensing data to be processed by CH. This basic static failure pattern can be modelled using Boolean *AND* gate between the two components.
- The other way to have the same failure is for the monitor to develop a fault; at a later time frame, the WSN A fails. Even while WSN B is operational, the system will also fail due to the inability of the MS to detect WSN A failure and activate WSN B. This dynamic failure is time-dependent and can be best modelled using a *PAND* gate.

Table 1 represents the dynamic failure analysis generated based on the IBD. The outcome of the dynamic failure analysis of the system will be factored into generating the system’s overall DFT.

Table 1. Dynamic Failure Behaviour Analysis of SFDS

Dynamic behaviour	Components	Appropriate Gate
Monitor system failure first and WSN A failure second	WSN A and monitor system	<i>PAND</i> gate

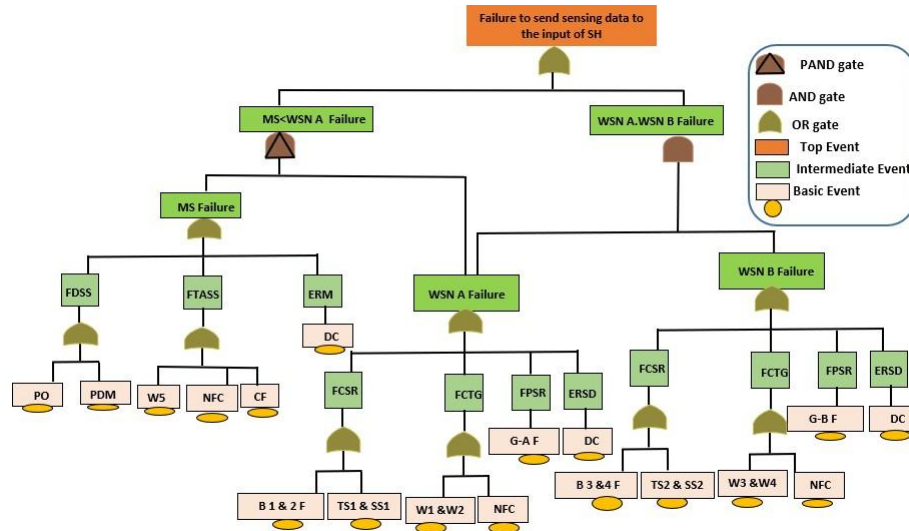


Fig.7. Dynamic Fault Tree Generated for the System

4.8 Overall DFT Generation

The DFT of the SFDS was finally developed based on the composition of the various FTs generated according to their instances in the IBD and the dynamic failure table generated. The mapping of the overall DFT was achieved based on the link between the source model developed in the MBSE and the executable DFT analysis model. The DFT generated from the three components used is shown in Fig. 7. The DFT represents the dynamic failure condition of three components selected in the IoT-enabled SFDS. The top undesired event of the selected component is the absence of reliable data from the sensing units to the input of the intelligent Hub. Based on the modelling, the intermediate events that directly contribute to the top events are failure to detect sensor status (FDSS), failure to actuate standby sensor (FTASS), erroneous monitor or sensor readings (ERM/ERSD), failure to capture sensor reading (FTSR), failure to communicate to the gateway (FCTG), failure to process sensor reading (FPSR) and erroneous sensor data (ERSD). Importantly, these intermediate events are linked not only by the basic Boolean *OR* and *AND* gates but also by the dynamic *PAND* gate, which depicts the dynamic failure behaviour of the system. Lastly, the basic events in the generated DFT are power outage (PO), physical damage of monitor system (PDM), wireless device failures (W 1-5), network failure condition (NFC), component failure (CF), battery failure (BF), temperature sensor failure (TS) and smoke sensor failure (SS).

5 Conclusion

Safety analysis of IoT-enabled systems is increasingly dynamic due to system complexity and design considerations. This paper proposes an MBSE approach for

practical safety analysis of IoT systems, either at a design stage or already existing systems, by relying on the vastness and expressiveness of UML/SysML libraries and their extension mechanisms. The proposed approach was based on the ability of the existing BDDs, IBDs, SMDs and *DaStep* stereotypes to represent an IoT system in an MBSE environment. This enables the development of a graphical model of the system, which was systematically mapped to develop a DFT model for safety analysis. We applied this approach to one of the safety-critical IoT applications, SFDS. In the hypothetical SFDS, we used our approach to generate the DFT of the selected part of the system. Our approach has simplified the safety analysis of IoT systems by helping the design process understand how various components and dynamic failures between components could culminate into system failure. This explains the root, secondary, and dynamic causes of system failure and erroneous states. Thus, enabling prioritisation in design considerations, component selection, and configuration will result in dependable IoT systems. This approach has some limitations, which include the relation of dynamic failure only between components, manual mapping of source model to formal analysable model and omission of quantitative analysis with the DFT generated—these open further research efforts, which we hope to address in our future work.

References

1. Abdulhamid, A., Kabir, S., Ghafir, I., Lei, C. (2023). An Overview of Safety and Security Analysis Frameworks for the Internet of Things. *Electronics* 12(14), 3086.
2. Xing, L. (2020). Reliability in Internet of Things: Current status and future perspectives. *IEEE Internet of Things Journal*, 7(8), 6704-6721.
3. Hussaini, A., Qian, C., Liao, W., Yu, W. (2022). A taxonomy of security and defense mechanisms in digital twins-based cyber-physical systems. In: *IEEE International Conferences on Internet of Things*. pp. 597–604. IEEE.
4. Kabir, S. (2021). Internet of Things and Safety Assurance of Cooperative Cyber-Physical Systems: Opportunities and Challenges. *IEEE Internet of Things Magazine*, 4(2), 74-78.
5. Kabir, S., Gope, P., Mohanty, S. P. (2022). A Security-enabled Safety Assurance Framework for IoT-based Smart Homes. *IEEE Transactions on Industry Applications*, 59(1), 6-14.
6. Kriaa, S., Pietre-Cambacedes, L., Bouissou, M., Halgand, Y. (2015). A survey of approaches combining safety and security for industrial control systems. *Reliability engineering & system safety*, 139, 156-178.
7. Abdulhamid, A., Kabir, S., Ghafir, I., Lei, C. (2022). Dependability of the Internet of Things: Current Status and Challenges. In: *2022 International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME)*. pp. 1-6. IEEE.
8. Kabir, S. (2017). An overview of fault tree analysis and its application in model based dependability analysis. *Expert Systems with Applications*, 77, 114-135.
9. Aslansefat, K., Kabir, S., Gheraibia, Y., Papadopoulos, Y. (2020). Dynamic fault tree analysis: state-of-the-art in modeling, analysis, and tools. In: *Reliability Management and Engineering*; CRC Press: Boca Raton, FL, USA. pp. 73–112.
10. Baklouti, A., Nguyen, N., Mhenni, F., Choley, J. Y., Mlika, A. (2019). Dynamic fault tree generation for safety-critical systems within a systems engineering approach. *IEEE Systems Journal*, 14(1), 1512-1522.

11. Kabir, S., Papadopoulos, Y., Walker, M., Parker, D., Aizpurua, J. I., Lampe, J., Rde, E. (2017). A model-based extension to HiP-HOPS for dynamic fault propagation studies. In: International Symposium on Model-Based Safety and Assessment. pp. 163-178. Springer.
12. Kabir, S., Walker, M., Papadopoulos, Y. (2018). Dynamic system safety analysis in HiP-HOPS with Petri Nets and Bayesian Networks. *Safety science*, 105, 55-70.
13. Alshboul, B., Petriu, D. C. (2018). Automatic derivation of fault tree models from SysML models for safety analysis. *Journal of Software Engineering and Applications*, 11(5), 204-222.
14. Bernardi, S., Gmez, A., Merseguer, J., Perez-Palacin, D., Requeno, J. I. (2022). DICE simulation: a tool for software performance assessment at the design stage. *Automated Software Engineering*, 29(1), 36.
15. de Andrade Melani, A. H., de Souza, G. F. M. (2020). Obtaining Fault Trees Through SysML Diagrams: A MBSE Approach for Reliability Analysis. In: 2020 Annual Reliability and Maintainability Symposium (RAMS). pp. 1-5. IEEE.
16. Baklouti, A., Nguyen, N., Mhenni, F., Choley, J. Y., Mlika, A. (2019). Improved safety analysis integration in a systems engineering approach. *Applied Sciences*, 9(6), 1246.
17. Wang, H., Zhong, D., Zhao, T., Ren, F. (2019). Integrating model checking with SysML in complex system safety analysis. *IEEE Access*, 7, 16561-16571.
18. Alshboul, B., Petriu, D. C. (2019). Pattern-based transformation of SysML models into fault tree models. In: Proceedings of the 29th Annual International Conference on Computer Science and Software Engineering, pp. 214-223.