ADDRESSING SAFETY ANALYSIS TECHNIQUE USING SysML (2.0) PARAMETRIC DIAGRAM

Maslina Binti Abdul Aziz 1, Noraini Binti Ibrahim 2

1 Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA Shah Alam, 40450 Selangor Darul Ehsan, Malaysia
2 Faculty of Computer Science and Information System, Universiti Teknologi Malaysia UTM Skudai, 81310 Johor, Malaysia
Email: maslina@tmsk.uitm.edu.my1, noraini@utm.my2

ABSTRACT

Complex safety critical systems are widely used in avionic, nuclear, medical, robotic, automation industries and defence applications. Safety critical system relies a lot on software, either it ensures a system's safety or can expose it by putting the system into a dangerous state. Therefore, when handling with safety critical systems, the first important step is to do hazards identification. Besides that, it requires a clear understanding of the whole software's role and interactions with the system. Identifying hazards is an important task in safety. It forms the basic foundation of the safety requirements, the system design and the system implementation. Hazard analysis needs to be carried out at the earliest stage of system development to avoid serious damage and accidents from happening. Identifying potential hazards helps to reduce overall development time, save project costs and efforts. This paper aims to validate the usability of the System Modeling Language (SysML) Parametric Diagram for safety evaluation. The described method is still an early attempt. The integration between the Parametric Diagram and safety analysis techniques provides a common language or platform for system engineers and customers to understand the whole system safety evaluation model and process. Therefore it provides a way to improve understanding on the safety properties of a safety critical system.

Keywords: SysML, Parametric Diagram, Safety Critical System, Hazards

1 INTRODUCTION

Having reliable software is the main key to many safety-critical systems. The software is important in order for these systems to function and meet their goals. The increasing number of such systems in areas such as avionic, nuclear, medical, robotics, automation industries and defence applications has indirectly effect additional capabilities in software. Future technological advances demands and consumer markets have caused the spread of safety-critical systems and it is expected to produce more safety-critical applications. Since they have a high rate of complexity, handling safety critical systems engage a large part of project development cost.

With growing complexity and emphasis on integration of software issues with system safety many classical safety analysis techniques become obsolete. There are a lot of issues especially on inconsistent and misinterpreted results especially regarding the design issues. One of the most common design issues is design integration. This is because each safety analysis technique has its own different types of notations. Different models caused significant problems such as inefficient communication, inability to share data, difficulty in tracing requirements, and duplicate work [14]. Results from each technique are different from one another and integrating them is often impossible.

This research studies modern safety analysis techniques that specifically support component-based systems that annotate systems components. These safety analysis techniques and the SysML Parametric Diagram will be briefly discussed in the next section. This paper is organised as follows. Section 2.0 describes in brief the context background of SysML. Accordingly, Section 3.0 outlines the overall concept of safety analysis techniques and the critical discussion on a simple case study of the techniques. Section 3.1 introduce s one of the safety analysis techniques, Failure Propagation and Transformation Notation (FPTN). In the following Section 2.2, discusses on another safety analysis techniques, Component Fault Tree (CFT). Initial results and findings are discussed in Section 4.0. Finally, Section 5.0 concludes the whole summary of this paper and presents the recommendations for for future improvements.
2 SYSML OVERVIEW

The Unified Modeling Language (UML 2.0) [2] has developed a new visual modeling language called Systems Modeling Language (SysML) [10, 17, 18, 19] specifically for system engineers. This new modeling language forms a bridge and integrates with other types of domain specific models, since it is also known as a general purpose language. Moreover, the SysML able to examine, identify, plan, and validate complex systems, intended for enhancing system quality, improve the ability to exchange systems engineering information amongst tools, and help to bridge the semantic gap between the systems and engineering disciplines [18]. These systems include hardware, software, data, personal, procedure and facility [10].

The SysML complements the UML 2.0 by reusing its components as much as possible without making any major changes to it. This idea is achievable when integrating the basic concept and features of the SysML new Requirements with Parametric diagram (figure 1). There are a number of advantages and benefits of parametric diagram. Among the benefits are they are modular, flexible and extendible to use.

The Parametric Diagram describes the constraints among the properties by representing them as with blocks [18]. It is also used to join together behaviors and structural models to determine the performance, reliability, and mass property models. Although there are various kinds of diagrams available, it does not integrate different kinds of diagram types, as one might do when one depicts the relationship between measures of effectiveness and various system properties.

Parametric Diagram supports modeling of constraints; constraint properties which bind their parameters together with the other properties of the internal block diagram. The parametric constraints are expressed using mathematical equations, statistical values or Boolean Algebraic formula using MathML or Object Constraint Language (OCL). The non-directional parametric constraints are used for block diagram without any notion of causality. The value will impact the value of others if changes exist.

An example of the Parametric Diagram and its relationships is shown in Figure 2. This example calculates the area and the parameter of a square. In the Composite Structure, the Square instance of the Size class, and the calculate instance of the range Object class are classified. In the Class section the Size and Area Object classes and their attributes are modeled. These individual objects are then linked together on the Parametric Diagram. Area is calculated as SideX x SideY, (both of which are attributes of the Range Object class), and Perimeter is equal to the sum of area and length.

3 SAFETY ANALYSIS TECHNIQUES

Accidents or failures in a safety critical system have serious consequences such as loss of life, significant property damage, or harm to the
environment. Even with a low chance, accidents are intolerable. In this case, identifying, controlling or eliminating hazards to acceptable levels is very demanding in order to avoid accidents [15]. Since the consequences of failures are very risky, the reliability, safety and fault tolerant must be the characters of safety-critical systems [18].

There are multiple types of safety analysis techniques. Each technique is diverged in terms of characteristics and the name itself. Different techniques may focus on different aspects and choosing the right aspects are very crucial in order to minimize and eliminate the possibility of hazards to occur. The natural disaster like wrong information in medication system and disable controller in railway system is considered as hazard in safety critical system [8]. The occurrence of hazard may result to various accidents or failures.

A system is considered fail-safe, when it affects system availability, not safety [24]. A system is reliable if it can perform the normal activities after the process of recover. A fault tolerant system is defined as a system that is able to produce the result even when it has to deal with failure in the system. It is important to ensure tasks or jobs are done in the safest manner possible, free from unacceptable risk of harm or danger. Different elements or components in a system have a different level hazard to people or equipment during their performance of task. Therefore, a tremendous safety measures need to be considered when constructing and handling safety critical systems.

All safety analysis techniques have their own capabilities and scheme. One technique may not applicable to all systems. A hybrid technique may be suit with the complexity of a system. The goal and objective of the project is the most entity in order to identify the usefulness of one particular technique. This research presents and briefly describes two primary safety analysis techniques. The safety analysis techniques are Failure Propagation Transformation Notation (FPTN) [4, 5, 6, 16] and Component Fault Tree (CFT) [6, 7 9, 12, 23]. Descriptions of these techniques are described in the next section.

3.1 Failure Propagation and Transformation Notation (FPTN)

The FPTN introduces a component concept that describes the any failure behavior of components in a hierarchical graphical notation [4, 5, 6, 14]. The uniqueness characteristic of FPTN as compared to the other conventional techniques is that, FPTN represents a system as a number of modules. The modules and some functional elements of a system are corresponded to each other. By maintaining this connection, the modules can annotate system components with failure propagation modules. Thus, the failure behavior can be detected by viewing the interconnected of modules with probability of failure mode.

The FPTN represents a system using three components which are a set of interconnected modules that contains program codes, failure modes and equations that expresses the output failure mode of the modules [6]. The FPTN analysis gives an excellent definition of the failure modes that needs to be considered and handled in a module [16].

<table>
<thead>
<tr>
<th>Notation</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>*\rightarrow</td>
<td>late source</td>
</tr>
<tr>
<td>Early \rightarrow*</td>
<td>sink</td>
</tr>
<tr>
<td>Omission \rightarrow o</td>
<td>propagate</td>
</tr>
<tr>
<td>Late \rightarrow stale value</td>
<td>transform</td>
</tr>
</tbody>
</table>

There are some notations that are used by FPTN (table 1). Table 1 illustrates the definition of notations that may be used in FPTN diagram. Different FPTN diagram may be used slightly different notation yet give the same definition or result [21].

FPTN also describes input and internal failure modes in the modules, figure 6 is an example of FPTN module that precisely focused on this matter. The combination of the failure mode of incoming input and internal fault of the component produce the failure mode of the outgoing output module [4]. These modules convey the information of the failures with each other and not by data flow. The FPTN modules transform the failures of one type into another different type by action. The failures can be classified into five types of reactions shown in table 2 [1].

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>tl</td>
<td>Too late</td>
</tr>
<tr>
<td>te</td>
<td>Reaction too early</td>
</tr>
<tr>
<td>ve</td>
<td>Value failure</td>
</tr>
<tr>
<td>c</td>
<td>Commission</td>
</tr>
<tr>
<td>o</td>
<td>Omission</td>
</tr>
</tbody>
</table>

In this next example (figure 3), the incoming failures are Failure1: o, Failure2: c, Failure3: e, Failure4: v and Failure5: l and the outgoing failure message is, Out: M. The transformation of failures is specified inside the module and specified with
sets of equations, for example (Out: M = Int1 || Failure1: o || Failure2: c || Failure3: e || Failure4: l || Failure5: v). Moreover, a component can also handle existing failures (for example Failure5:v) and determine its internal failure (for example Int1: o). Each module also contains an identifier (ID), a name and a critical Safety Integrity Level (SIL).

The FPTN modules indicate four activities which are the propagation failures, transformation, the generation of internal failure and the detection of the failures in the component. FPTN considers all possible failures from the environment that can affect the components and the component’s failures that might influenced the environment.

The FPTN modules indicate four activities which are the propagation failures, transformation, the generation of internal failure and the detection of the failures in the component. FPTN considers all possible failures from the environment that can affect the components and the component’s failures that might influenced the environment.

### 3.2 COMPONENT FAULT TREE (CFT) APPROACH

Component Fault Trees (CFT) [6, 7, 9, 12, 23] is an extended version of the approach taken by Fault Trees (FT). The evolution of CFT is due to limited function of FT in breaking up the mechanism into independent subtrees [12]. The CFT is a technique that analyzes inter-component failure flows, which can be used to perform quantitative analysis of the system-level failure [7].

The CFTs only need to be modeled once for a component type but can be reused as often as needed. A CFT enables the internal failure flow of the components to be modeled and analyzed. Each component can be treated and stored separately since it has independent subtrees. Ports are used to describe failures and act as an interface to the environment in which the component operates.

The subtree components represent the system’s real components which are recursively refined into sub-components. The sub-components model the internal failure propagation paths that permit the internal flow failure to be analyzed using a set of Boolean functions [12]. The subcomponents in the CFTs cannot be represented in the traditional

FT because each event is statistically independent towards each other.

![Figure 3. An abstract of FPTN Module](image)

![Figure 4. An abstract of CFT Module](image)

### 4 RESULTS AND DISCUSSION

#### 4.1 The Comparisons of Safety Analysis Techniques

Safety is beyond correctness. In this case, both FTPN and CFT are the techniques that focused on identifying and measuring failure that could be triggered hazard. Hazards are predictable and what ever that can be predicted can also be eliminated or controlled. FTPN and CFT are similar in terms of their way of expressing the characteristics of a system. They have a common way of describing the failure propagation of individual components (failure propagation and data flow). Therefore, they are compatible and exchangeable as safety notation between each other.

FTPN is concentrated on the failure behaviour rather than to correct the behaviour itself [21]. Each component failure propagation models are combined to have a model for the complete system. Based on the system level failure probabilities, hazard probabilities can be calculated using the UWG tool which was developed by the ESSAREL project [3]. At the end of the process, the safety engineer can check the probability of hazard. If the system does not satisfy the requirement, the system must be revised and additional safety
mechanisms (e.g. watchdogs) need to be included. This letting the FTPN technique to repeatedly improve the module until the most suitable result is identified.

The Fault Tree (FT) however is basically a hierarchical breakdown structure that connects failures starting from the root to other parts of the tree. It is a quantitative safety analysis technique that uses a graphical representation to model causal chain leading to failures. Among the features of the traditional FTs are that they are compostable and each entity is not reusable. It is impossible to utilize the components of the FTs for later integration with the whole system model. Each subtree is related, influenced each other and cannot be handled separately. Therefore the concept of having independent subtrees is neither applicable nor practical. They are combinatorial models and thus cannot consider state dependencies or temporal order of events [11].

The other drawback of the FT is the size of the generated fault trees of a complex system. Usually the modularity of the complex systems should be reflected by the FTs. It is not possible to use FTs because statistical independence of the causes contained in the subtrees have resulted in incorrect calculation of reliability characteristic [12]. One of the solutions to this problem is the use of repeated events that mark multiple occurrences of causes; however, the side effect of this approach is that the tree tends to grow larger and confusing.

4.2 Case Study Validation

The use of Parametric Diagram and two other safety analysis techniques described in previous section is explained with the help of a case study. The braking system context diagram is shown in Figure 5.

![Figure 5. CAB system context diagram [20]](image)

An example of automobile braking system case study, The Computer Assisted Braking System (CABS) developed at University of York [20]. This case study models the concept of an automatic braking system deployed in modern cars. The main function of CABS is to enhance traditional vehicle braking system performance by providing a better braking assistance (computerized system) to the existing traditional brakes (mechanical system).

The CABS has three additional functions as compared to the traditional braking system by providing: Anti-lock Braking (ABS), Emergency Stop Detector (ESD) and Load-Compensator Braking (LCB). The ABS function is to keep normal brakes from locking up. It is designed to improve steerability and avoid jammed wheels. It detects the onset of wheel lock-up which is useful in situation where the wheels skid or spin-out especially on slippery road. It works effectively in emergency situations when the brakes may not engage when required.

The CAB system architecture design proposed was modified from the original CAB system architecture for better understanding and ease of use. The changes were made in the controller component, three processors (ABS, ESD and LCB) were treated separately and a modified output module. The CAB system (Figure 5) is divided into four main components and five sub-components. The components are:

a) Wheel Rotation Sensors (WRS)
b) Axle Load Sensors (ALS)
c) Brake Pedal Sensors (BPS)
d) Controller
   i. Anti-lock Processor (ABS)
   ii. Emergency Stop Detector Processor (ESD)
   iii. Load-compensator Braking Processor (LCB)
   iv. Dual CAN Bus
   v. Output Module

The corresponding figures illustrated below are examples of the use of Parametric Diagram, FTPN and CFT. These examples only represent one part of the braking system, the Anti Lock System (ABS) component. The respective techniques translate the fault propagation models. Each component is modeled separately in order to have better understandings about Parametric Diagrams.

i. Parametric Diagram with ABS_ModifierValue.o
   | ABS_ModifierValue.o = Computer data storage failure (controller) v (fr.WRS_1.o v fl.WRS_2.o Λ rr.WRS_3.o v rl.WRS_4.o) |
Figure 6 above illustrates the use of Parametric Diagram representing one of the components of the CABS, Anti Lock Braking Processor (ABS). It indicates one of the relationships between the anti-Lock Processor (output failure) and the faults either generated from the sensors (fr.WRS.o, fl.WRS.o, rr.WRS.o and rl.WRS.o) or the ABS internal faults (computer data storage failure (controller)). They are related by 2 OR gates. This means that any failure caused by one of the 4 WRS sensors combined with the internal failure can lead to output faults. Based on this figure, it showed that Parametric Diagram is non-directional and have no notion of causality. Therefore, the information flows between them can be misinterpreted. It can only be understood by people, who are familiar with the system. The Figure 7 below represents the ABS component using FPTN.

ii. FPTN for ABS_ModifierValue.o

Similar with the Parametric Diagram above, the FPTN has incoming input failures from the WRS sensors (fr.WRS.o, fl.WRS.o, rr.WRS.o and rl.WRS.o), the outgoing failure message is, ABS_ModifierValue.o and the internal fault is computer data storage failure (controller).

The Parametric Diagram of ABS_ModifiedValue.o is translated to component fault tree (CFT) diagram (figure 8). Similar to the Parametric Diagram, the four sources of input failures (fr.WRS.o, fl.WRS.o, rr.WRS.o and rl.WRS.o) and an internal fault (computer storage failure (controller)) are logically related to the outgoing failures. They are related with OR Gates. Which means that, either one of the input failures OR the internal fault can cause the ABS_ModifierValue.o (output failure) to happen.

iii. CFT for ABS_ModifierValue.o
translated to the CFTs. Drawing from these initial results, it illustrated how Parametric Diagram can be used as an alternative to represent failure propagation models and helped to overcome some of the problems with the classical safety analysis techniques.

The Parametric Diagram, FPTN and CFT are similar in terms of their way of expressing the characteristics of a system. They have a common way of describing the failure propagation of individual components (failure propagation and data flow). Therefore, they are compatible and exchangeable as safety notation between each other.

5 CONCLUSION AND FUTURE WORKS

This paper was mainly about exploring and evaluating whether it is practical and beneficial to use the SysML Parametric Diagram [11, 17, 18, 19] as failure propagation notation for safety critical system. Each safety analysis technique (FPTN and CFT) and the Parametric Diagram are capable of representing different types of notations specifically the failure propagation models.

Inconsistent designs and different representations between various safety studies are among the problems when employing with classical safety analysis techniques. Based on this research, it was proven that the solution for linking the initial results and findings of different types of notations to represent the failure propagation models can be done by using Parametric Diagrams. Parametric Diagram acts as a “bridge” to assess the quality of the system. It can also be used as an alternative standard notation or “common language” that can be understood by system engineers and other stakeholders who are familiar with SysML. The choices of which types of diagram representation from which safety analysis technique to use, will depend on the system requirements to be modeled.

Based on the initial results of this research, it is proven that there are some downsides and disadvantages of the proposed method. Firstly, the size of the parametric diagrams can become too big. The combinations of parametric diagrams will make the diagrams grow bigger and difficult to read. The more complex the system would be, the bigger they get. Secondly, the diagram produced is very complex and can only be assessed or translated by expert, skilled and experienced knowledge of a team. Thirdly, the downside of the parametric constraints is that they are non-directional so they have no notion of causality and therefore, it is hard to understand the overall system.

The overview of this research highlighted how SysML parametric diagrams can be integrated with two safety analysis techniques; FPTN and CFT. The described method is still an early attempt for safety evaluation and it needs more in-depth research. There are still a lot of the SysML Parametric Diagram features that need to be explored. This research should be expanded by integrating more safety analysis techniques with the SysML Parametric Diagram. Moreover, the application of the parametric diagram required further evaluation and practicability. It can be done by applying it to safety critical systems or a more interactive real-time system to take full advantage of it.

REFERENCES


