Software Safety Analysis with UML-Based SRBD and Fuzzy VIKOR- Based FMEA

Sh. Oveisi[1,2] and M. A. Farsi[1*]

1. *Aerospace Research Institute, Ministry of Science, Research and Technology, Tehran, IRAN*

2. *Department of Algorithms and Computation, School of Engineering Sciences, University of Tehran, Tehran, Iran*

Abstract

Software often controls the behavior of mechanical and electrical systems, as well as interactions among their components in cyber-physical systems (CPS). The risks in CPS systems could result in losing tools, features, performance, and even life. Therefore, safety analysis for software in these systems is a highly critical and serious issue. The use of reliability block diagram is a method for checking the safety and reliability of systems. A reliability block diagram is a diagrammatic method for showing how component reliability contributes to the success or failure of a complex system. In this paper, a method for generating RBDs is presented analysis and demonstration of this method capability to evaluation of a software safety by use-case analysis, use-case diagram review, and use-case specification. Then, a Fuzzy VIKOR-based FMEA is used for further evaluation due to the presence of uncertain data. Finally, it is applied to a real CPS.

Keywords: Software safety, SRBD, UML, Fuzzy VIKOR

Nomenclature and Abbreviations

CPS	Cyber Physical System
IBR	Inquiry Board Report
RBD	Reliability Block Diagram
MCDM	Multi-Criteria Decision Making
UML	Unified Modeling Language
UC	Use -Case
SRBD	Software Reliability Block Diagram
DCU	Data and Command Unit
SFTA	Software Fault Tree Analysis
OOD	Object Oriented Design
AHP	Analytic Hierarchy Process
SFMEA	Software Failure mode and effects analysis
RPN	Risk Priority Number
FMEA	Failure mode and effects analysis
CCF	Common Cause Failure

1. Introduction

CPSs are integrations of computation, networking, and physical processes. Embedded computers and networks monitor and control the physical processes, with feedback loops where physical processes affect computations and vice versa. Examples of CPS include aerospace systems, transportation vehicles and intelligent highways, robotic systems, intelligent environments, and spaces, etc. [1-2].

The discovery of flaws has become more difficult with the increasing complexity of CPSs. Software is the cornerstone of CPS. The complexity of these software with millions of lines of code can cause dangerous consequences regarding the failure of these software [3]. Final flaws in requirements, design, or execution of software can lead to unpredictable events at the integration level of software [4]. for example, Arian 5 after a short time of flight due to a software fault failed that cost a lot [5].

There are several ways to establish safety in hardware and software components. RBD is an approach to establish safety in software components, which can be implemented at different levels of software development for various purposes.

A RBD performs the system reliability and availability analyses on large and complex systems using block diagrams to show network relationships [6]. The structure of the reliability block diagram defines the logical interaction of failures within a system that are required to sustain system operation [7].

In order to identify the causes, prioritizing and eliminating failures from the system or software requires another method called FMEA.

FMEA is a popular and useful approach applied to examine potential failures in products, designs, processes, and services [8-9]. As a vital index, the RPN can determine the risk priorities of failure modes by some risk factors such as occurrence (O), severity (S), and detection (D) [10-11].

[*] Corresponding author. Tel.: +9821 88366030
E-mail address: farsi@ari.ac.ir

In order to prioritize the risks and select the most important and most effective risks, one of the MCDMs selected using fuzzy theory.

Fuzzy set theory is a way of addressing vague concepts, which provides a means for representing uncertainty involved in the real situation. On the other side, the VIKOR method is a recently developed MCDM method, which focuses on ranking and selecting from a set of alternatives in the presence of conflicting criteria and on proposing compromise solution. For selecting the most serious failure modes, an extended VIKOR method was used to determine the risk priorities of the failure modes that have been identified. As a result, a fuzzy FMEA model based on fuzzy set theory and VIKOR method was proposed for the prioritization of failure modes, specifically intended to address some limitations of the traditional FMEA [11].

Requirements analysis during all stages of software development plays the most important role in determining the safety of the whole software. If problems and errors leading to software failure are identified and their risks are reduced at this stage, then, the level of risk and system failure are significantly decreases during later software development processes [28]. A modeling language that is used in different phases of software development, including the requirements phase, is UML for which solving complex, the object-oriented problems in the field of software engineering which is standardized and commonly used by the software development community. UML applies a number of diagrams and views to describe software systems.

The identification and development of use-cases (UC) is highly critical in the requirement analysis phase of such systems given the rough requirements of a CPS system [12-13]. To detect the faults in CPS during an early stage of the design, we need a method to analyze the RBD for the developed UCs presented in requirements' engineering phase to detect the fault and prevent it as soon as possible. For this purpose, a method, namely SRBD, is presented in this paper that is capable of producing SRBD for UCs developed for CPS.

To evaluate the SRBD, it was applied on UCs of the Data and Command Unit. The Data and Command Unit is a main subsystem in aerospace systems, which issues the start-of-movement signal that is the basis of all subsequent activities of this type of systems. The essential system operation software is responsible for issuing commands such as sending and receiving signals between internal parts of the system, setting micro timer, and so forth. Unforeseeable interactions between software, hardware, and the environment can lead to potentially hazardous conditions. Moreover, to reduce the effect of failures using VIKOR- Based Software FMEA.

The rest of this paper is organized as follows: After the introduction, research background is presented in Sec. 2. Section 3 introduces proposed approach and then RPN and VIKOR based SFMEA are described. Finally, Risk analysis of highly risk events are done.

2. Research Background

Software safety techniques play an important role in software development and are a valuable factor in the life cycle. Several studies have been so far conducted in different phases of software development cycle to increase the safety and reliability of software. A number of safety -related works using the mentioned methods are briefly cited below.

A manual four-step solution has been presented to integrate SFMEA and SFTA for the analytical process of use-case based requirements. In this approach, the UML use-case model is translated to a software fault tree for the analysis of safety based on system behavior. A text-based use-case model, known as use-case specifications, is used in this approach to produce the SFMEA to reduce the fault effects and results [14]. Vyas and Mittal propose another approach to extract safety requirements in a manual systematic form of use-case requirements. SFTA has been demonstrated according to use-case then the authors have validated the results of their approach using a real case study in elevator control system [15]. An effective method has been suggested to organize the information of fault tree and reuse SFTA information to produce the software fault tree [16]. Four different steps have been designed for this method. In the first step, information of software fault tree is described by a semiformal method in the form of elements such as nodes, relations, target functions, and target software modules. Then, a knowledge base is constructed for information of software fault tree. For this purpose, different attributes of each node are considered. Finally, a reusable fault tree is automatically produced from the knowledge base using compliance between the texts with intelligent relations. This approach has been applied in the aerospace software systems.

Oveisi and Ravanmehr, after reviewing the major techniques of software reliability and safety in CPS, a software fault tree analysis (SFTA)-based approach is presented for analysis of operational use-cases (UC) in a CPS system [17].

A novel approach for FMEA based on combination weighting and fuzzy VIKOR method is presented by Hu-chen et al.[18].Integration of fuzzy analytic hierarchy process (AHP) and entropy method is applied for risk factor weighting in this proposed approach. The risk priorities of the identified failure modes are obtained through next steps based on fuzzy VIKOR method.

Safari et al.[19].,instead of calculating RPN-prioritizes Enterprise architecturerisk factors with fuzzy VIKOR. As regards using linguistic variables, the fuzzy approach is used to allow experts to use linguistic variables. The proposed method was used for evaluating twenty Enterprise architecturerisk factors, which integrates knowledge and experience acquired from professional experts.

Using UML diagram, Rajput and choury created RBD and this RBD use to evaluate the components of the software[20].

3. Proposed Approach

Although several methods have been suggested to increase safety and guarantee in the software development process, less attention has been paid to the evaluation of a method based on these inaccurate data given the uncertainty of the events leading to failures before a software is used. Accordingly, the workflow method proposed to enhance the reliability, and reduce the risk is presented in Fig. 1. The issues discussed in the proposed method are then evaluated below.

3.1. Work flow of proposed method

Automatically SRBD generation from use-case specifications is the most important section of the proposed method. The system and software should be carefully examined, and the specification table is obtained by evaluating use-case specifications and use-case diagrams. After automatic generation of SRBD, due to the existence of uncertain data, fuzzy VIKOR-based FMEA is applied to increase accuracy in the determination of the system reliability.

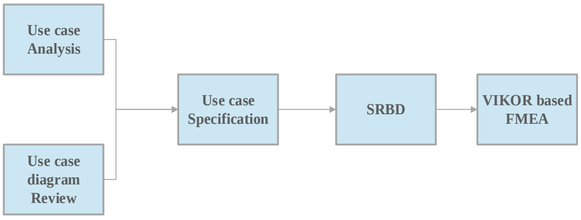

Fig.1. Workflow of the proposed method

3.2. SRBD using UML modeling

The main purpose of using SRBD during software development is to identify the weaknesses in requirements specifications. To this end, the weak requirements are changed or other requirements will be added. All conditions having a direct impact on the safety of the system are identified. When requirements with safety considerations are identified, these requirements will be tracked throughout the lifecycle development [1].

During software design, using UML, the software is displayed in the form of a number of modules in which the interfaces, inputs and outputs are specified. Application of SRBD at this step enables us to Identify modules (objects, methods or functions) which can directly affect the system's safety. It should be noted that RBD generation for the system is a much more efficient choice at specifications and design phase than their generation in the implementation phase, because RBD generation in the latter phase is a heavy task demanding intensive work.

The object-oriented design (OOD) can be selected as an approach to use the SRBD at the design level. There are two main reasons for choosing OOD: 1) Many recent software designs use OOD and software are implemented using OO languages 2) Recently, a large number of OODs use UML which is standardized by the software community and is commonly used by this community. UML takes advantage a number of viewpoints and diagrams to describe software systems.

3.3. Review of use-cases: events extraction

In this paper, the use-case models provide both functional requirements and behavioral analysis. In fact, this model has been customized to be suitable as an input to SRBD. For this purpose, related eventsare extracted for each use-case using use-case diagrams. Here, the events are specified as essential events and are displayed as $\{P_1, P_2, ..., P_i\}$. The main flow of operations is shown as $\{M_1, M_2, …., M_j\}$ and the alternative flow of operations depending on each event is shown as $\{S_1, S_2, …., S_n\}$.

Table 1. Use-case template

Use-case <No>	<Use-case Name> Provide a short meaning full name
Preconditions $\{P_1,P_2,….,P_i\}$	Preconditions indicate circumstances that must be true prior to the invocation of any event of the use-case. Alternative of preconditions are possible. They are ordered by numbering with letters $(P_1,P_2,…,P_i)$

Main Flow $\{M_1,M_2,..M_j\}$	Main flow indicates the intended functionality of the given use-case$\{M_1,M_2,\ldots,M_j\}$ and indicates Dependence As is shown below: $P_1{:}M_1,\ldots\ldots\ldots,P_1{:}M_j$ $P_2{:}M_1,\ldots\ldots\ldots,P_1{:}Mj$ … $P_i{:}M_1,\ldots\ldots\ldots,P_i{:}M_j$
Alternative Flow $\{S_1,S_2,\ldots,S_n\}$	Alternate flows will always indicate the exceptional scenarios$\{S_1,S_2,\ldots,S_n\}$ and indicate Dependence As is shown below: $P_1{:}M_1{:}S_1,\ldots\ldots\ldots,P_1{:}M_1{:}S_n$ $P_1{:}M_2{:}S_1,\ldots\ldots\ldots,P_1{:}M_2{:}S_n$ … $P_i{:}\ M_1{:}S_1,\ldots\ldots\ldots,P_i{:}M_1{:}S_n$ …. $P_i{:}M_j{:}S_1,\ldots\ldots\ldots,P_i{:}M_j{:}S_n$

We have used the method proposed by Oveisi and Ravanmehr [17] to obtain preconditions, main flow, and alternative flow.

3.4 Characteristic table conversion into SRBD

Considering the format of the characteristic table provided in Table 1, the characteristic table conversion into SRBD is completed as follows:) If the failure of all alternative flows used for one main flow leads to failure of that main flow, they can be connected in series.

2) If the failure of one alternative flow used for one main flow results in failure of that main flow, they can be connected in parallel.
3) For main flows as presented in Table 1, it is as follows: we have different main flows and alternative flows separately.

Thus, if the failure of one precondition leads to the inability of our sub-system, they will be connected in series. However, when the failure of all preconditions results in deactivation of our subsystem, all alternative flows of main different flows will be connected in parallel.

4. RPN

RPN is a risk assessment method in FMEA. In this method, three factors of Occurrence (O), Severity (S), and Detection (D) are used, the product of which is equal to RPN given the independence of these factors. These three factors are traditionally divided into ten levels, which are shown in Tables 2, 3, and 4.

$$RPN = D * S * O \qquad (1)$$

Table 2. Severity of Failures [21]

Severity criteria	Value	Detection
The effect is not noticed by the user.	1	None
Very slight effect noticed by the user. Does not annoy or inconvenience the user.	2	Very minor
Slight effect that causes user annoyance, but they do not seek service.	3	Minor
Slight effect, user may return product for service.	4	Low
Moderate effect, user requires immediate service.	5	Moderate
Significant effect, causes user; dissatisfaction.	6	Significant
Major effect, system may not be operable; elicits user complaint; may cause injury	7	Major
Extreme effect, system is inoperable and a safety problem, may cause service injury	8	Extreme
Critical effect, complete system shutdown, safety risk	9	Serious
Hazardous, failure occurs without warning.	10	Hazardous

Table 3. Failure Detection [21]

Criteria: Likelihood of Detection by Design Control	Detection	Rank
Design control will almost certainly detect a potential cause of failure or subsequent failure mode	Almost certain	1
Very high chance the design control will detect a potential cause of failure or subsequent failure mode	Very high	2
High chance the design control will detect a potential cause of failure or subsequent failure mode	High	3
Moderately high chance the design control will detect a potential cause of failure or subsequent failure mode	Moderately high	4
Moderate chance the design control will detect a potential cause of failure or subsequent failure mode	Moderate	5
Low chance the design control will detect a potential cause of failure or subsequent failure mode	Low	6
Very low chance the design control will detect a potential cause of failure or subsequent failure mode	Very low	7
Remote chance the design control will detect a potential cause of failure or subsequent failure mode	Remote	8
Very remote chance the design control will detect a 9 potential cause of failure or subsequent failure mode	Very remote	9
Design control does not detect a potential cause of failure 10 or subsequent failure mode, or there is no design control	Absolute uncertainty	10

Table 4. Traditional FMEA scale for occurrence [22]

Probability of Failure	Possible Failure Rates	Rank
Extremely high: Failure almost inevitable	≥in 2	10
Very high	1 in 3	9
Repeated failures	1 in 8	8
High	1 in 20	7
Moderately high	1 in 80	6
Moderate	1 in 400	5
Relatively low	1 in 2000	4
Low	1 in 15,000	3
Remote	1 in 150,000	2
Nearly impossible	1 in 1,500,000	1

4.1. Fuzzy VIKOR- Based RPN

Failure Mode and Effects Analysis (FMEA) is an analysis method of potential failure in products or processes, which is used in many quality management systems. FMEA is an important issue in determining the risk priorities of failure scenarios. In classical FMEA method, the risk priorities of failure modes are determined by risk priority numbers (RPNs) through multiplying risk factors such as severity (S), occurrence (O), and the probability of detection (D). However, definite RPNs have been criticized by many scholars and experts because of their shortcomings and disadvantages, so that significant efforts have been made in FMEA literature to address these shortcomings [23] [24].

In this paper, we used FMEA which is a powerful tool for risks evaluation. in traditional FMEA, risk priority number(RPN), has been calculated by multiplication of three criteria, the severity of the traditional FMEA, in this paper, instead of calculating RPN-prioritizes risk factors with fuzzy Vikor.

4.1. Fuzzy VIKOR- Based RPN

This method focuses on ranking and selecting from a set of alternatives, and determines compromised solutions for a problem with conflicting criteria, which can help the decision makers to reach a final decision [25]. As regards using linguistic variables, the Fuzzy approach is used to allow experts to use linguistic variables. The fuzzy VIKOR method has been developed to determine the compromise solution of the fuzzy multicriteria problem

$$mco_j\{(\tilde{f}_{ij}(A_j), j = 1, \dots., J), i = 1, \dots., n\}$$

Where: J is the number of feasible alternatives; $A_j = \{x_1, x_2, \dots\}$ is the jth alternative obtained (generated) with certain values of system variables x; f_{ij} is the value of the ith criterion function for the alternativeA_j; n is the number of criteria;mco denotes the operator of a multicriteria decision making procedure for selecting the best (compromise) alternative in multicriteria sense. Alternative can be generated and their feasibility can be tested by mathematical models (determining variables x), physical models, and/or by experiments on the existing system or other similar systems. Constraints are seen as high-priority objectives, which must be satisfied in the alternatives generating process. In this paper, we assume the alternatives are evaluated by the triangular fuzzy numbers$\tilde{f}_{ij} = (l_{ij}, m_{ij}, r_{ij}), i = 1, \dots, n,\ j = 1, \dots, J$. The set of criteria representing benefits (good effects) is denoted byl^b, and a set l^c for costs. Here $|l^b \cup l^c = n|$ where $|.|$ denotes a cardinal number.

The ranking algorithm VIKOR has the following steps [11]:

Determine the best (aspired/desired levels) $\tilde{f}_i^* = (l_i^*, m_i^*, r_i^*)$ and the worst (tolerable/worse levels) $\tilde{f}_i^\circ = (l_i^\circ, m_i^\circ, r_i^\circ)$ values of all criterion functions $i = 1,2, \dots, n$.

$$\tilde{f}_i^* = MAX_j\tilde{f}_{ij},\ \tilde{f}_i^\circ = MIN_j\tilde{f}_{ij},\ for\ i\ \in\ l^b;$$

$$\tilde{f}_i^* = MIN_j\tilde{f}_{ij},\ \tilde{f}_i^\circ = MAX_j\tilde{f}_{ij},\ for\ i\ \in\ l^C.$$

Compute the gaps (normalized fuzzy difference) $\tilde{d}_{ij}, j = 1, \dots, J, i = 1, \dots, n$:

$$\tilde{d}_{ij} = (\tilde{f}_i^* \ominus \tilde{f}_{ij})/(r_i^* - l_i^\circ)\ \ for\ i \in l^b \qquad (2)$$

$$\tilde{d}_{ij} = (\tilde{f}_{ij} \ominus \tilde{f}_i^*)/(r_i^\circ - l_i^*)\ \ for\ i \in l^c$$

Compute$\tilde{S}_j = (S_j^l, S_j^m, S_j^r)$

and$\tilde{R}_j = (R_j^l, R_j^m, R_j^r), j = 1,2, \dots, J$,by the relations

$$\tilde{S}_j = \sum_{i=1}^n \oplus(\widetilde{w}_i \otimes \tilde{d}_{ij}) \qquad (3)$$

$$\tilde{R}_j = MAX_I(\widetilde{w}_i \otimes \tilde{d}_{ij}) \qquad (4)$$

Where $\tilde{S}$ is a fuzzy weighted sum, $\tilde{R}$ is a fuzzy operator MAX, $\widetilde{w}_i$are the weights of criteria, expressing the DM's preference as the relative importance of the criteria.

Compute the values$\tilde{Q}_j = (Q_j^l, Q_j^m, Q_j^r), j = 1,2, \dots, J$, by the relation,

$$\tilde{Q}_j v\, (\tilde{S}_j \ominus \tilde{S}^*)/(S^{\circ r} - S^{*l}) \oplus (1 - v)\, (\tilde{R}_j \ominus \tilde{R}^*)/(R^{\circ r} - R^{*l}) \qquad (5)$$

Where:

$$\widetilde{S}^* = MIN\tilde{S}_j, S^{\circ r} = max_j S_j^r, \tilde{R}^* = MIN\tilde{R}_j, R^{\circ r} = max_j R_j^r,$$

and ν is introduced as a weight for the strategy of "the majority of criteria" (or "the maximum group utility"), whereas 1-ν is the weight of the individual regret. These strategies could be compromised by ν=0.5 and here ν is modified as $\nu = (n+1)/2n$ from $(\nu + 0.5\,(n-1)/n = 1)$ since the criterion (1 of n) related to R is included in S, too. The best values of S and R are denoted by $\tilde{S}^*$and $\tilde{R}^*$, respectively.

"Core" ranking

Rank the alternative by sorting the core values $Q_j^m, j = 1,2,\dots,J$, in decreasing order. The obtained ordering is denoted by$\{A\}_{Q^m}$.

"Fuzzy" ranking

The j$_{th}$ ranking position in $\{A\}_{Q^m}$ of an alternative $A^{(j)}, j = 1,2,\dots,J$, is confirmed if $MIN_{k\in J^j}\tilde{Q}^{(k)} = \tilde{Q}^{(j)}$, where$J^j = \{j, j+1,\dots,J\}$ and $\tilde{Q}^{(k)}$ is the fuzzy merit for the alternative $A^{(k)}$ at the kth position in $\{A\}_{Q^m}$. Confirmed ordering represents "exact" fuzzy ranking $\{A\}_{\tilde{Q}}$,although the set $\{A\}_{\tilde{Q}}$ could not be complete ordering (it may be partially ranking).

Defuzzification of $\tilde{S}_j,\ \tilde{R}_j,\ \widetilde{Q}_j,\ j = 1,2,\dots,J$, by the relations

$$Crisp(\widetilde{N}) = (2m + l + r)/4 \qquad (6)$$

Here the defuzzification method "2nd weighted mean" is applied to convert a fuzzy number into crisp score.

Rank the alternatives, sorting by the crisp values S, R and Q in decreasing order. The results are three ranking lists$\{A\}_S, \{A\}_R, \{A\}_Q$.

Propose as a compromise solution the alternative($A^{(1)}$) which is the best ranked by the measure Q (in $\{A\}_Q$) if the following two conditions are satisfied:

C1. "Acceptable Advantage": $Adv \geq DQ$

Where:

$Adv = \left[Q(A^2) - Q(A^{(1)})\right]/\left[Q(A^J) - Q(A^{(1)})\right]$ is the advantage rate of the alternative $A^{(1)}$ ranked first, $A^{(2)}$is the alternative with second position in $\{A\}_Q$ and the thresh old

$DQ = 1/J - 1.$

C2: "Acceptable Stability in decision making":

The alternative $A^{(1)}$ must also be the best ranked by S or/and R.

If one of the conditions is not satisfied, then a set of compromise solutions is proposed, which consists of:

Alternatives $A^{(1)}$and $A^{(2)}$ if only the condition C2 is not satisfied, orAlternatives $A^{(1)}, A^{(2)}, \dots, A^{(M)}$ if the condition C1 is not satisfied;$A^{(M)}$ is determined by the relation $Q\left(A^{(M)}\right) - Q(A^{(1)}) < DQ$ for maximum M (the positions of these alternatives are "in closeness").

Determine crisp tradeoffs, $tr_{ik} = (D_i w_k)/(D_k w_i)\,, K \neq I, K = 1,\dots,n$

where tr_{ik} is the number of units of the ith criterion evaluated the same as one unit of the k$_{th}$ criterion; $D_i = r_i^* - l_i^\circ$ for $i \in l^b$for $D_i = r_i^\circ - l_i^*$, and $i \in l^c$ obtained by defuzzification. The index is given by the VIKOR user. The VIKOR method introduces these trade-offs as a result of normalization used in Eq. (2) for operations in (3) and (4)

5. Case Study

We applied the results of our approach to a part of a real CPS known as Data acquisition and Command Unit, the architecture of which is shown in Figure 2.

The most important goal of the Data acquisition and Command Unit is the timely release of commands for separation of a part, engine, and parachutes based on the simulated time and height. To begin working, this section needs to detect the start of the movement, and in fact, it must receive the Start signal. Start signal, which results from the simultaneous cut of cord and compression of mass and spring switch, is a command to start the operations of the two system processors, which use data from pressure sensors and timeline of their internal timers to perform their operations.

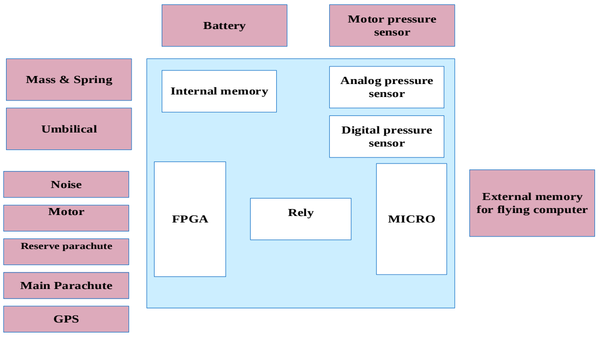

Fig. 2. The general architecture of Data acquisition and Command Unit

5.1. Evaluation of the designed software

Data acquisition and Command Unit is one of the most important subsystems of aerospace systems. As discussed earlier, since the use of code-level SRBDs is a difficult task, it has been attempted to perform analyses to establish software safety from the early phases of software lifecycle for this set.

Our focus in this paper is to analyze the design of software used in Data acquisition and Command Unit. Accordingly, after reviewing the Use-cases, the use-case diagram was obtained both in general and for all subsets. Then, the use-case specifications were written for them. The method proposed to generate SRBD is based on the approach proposed by Oveisi and Ravanmehr [17]. After generation of SRBD for further evaluation, RPN of baseline events was obtained using fuzzy VIKOR-based FMEA. In this paper, this case is called Analog to Digital, which transforms the sensor's analog data into digital data. Therefore, in Figure 3, use-case analog to digital is displayed, and its use-case specifications are shown in Table 5. After reviewing the specification table and use-case, SRBD of each event was obtained by examining the events shown in Table 6 (Fig. 4), and its reliability was determined using the equation.

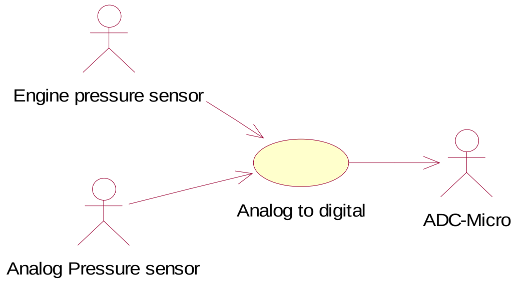

Fig.3. Use-case diagram of analog to digital

Table 5. Use-case specifications of evaluate visibility

Use-case <No>	Evaluate visibility
Preconditions $\{P_1,P_2,\ldots,P_i\}$	P_1: Analog pressure sensor is not ready P2: Micro is not ready
MainFlow $\{M_1,M_2,..M_j\}$	P_1:M_1: Failure in sending true data to Micro P_2:M_1: Failure in AD mux P2:M2: reading data from ADC port P2:M3: Failure in receiving data from analog pressure sensor by ADC

Table 6. Events of analog to digital

Event	Event name
Failure in sending data to Micro	X1
Failure in AD MUX	X2
Failure in reading data from ADC port	X3
Failure in Receiving data from Analog Pressure Sensor by ADC	X4
Analog Pressure sensor is not ready	I1
Micro is not ready	I2
Failure in Receiving data from Analog Pressure Sensor	T

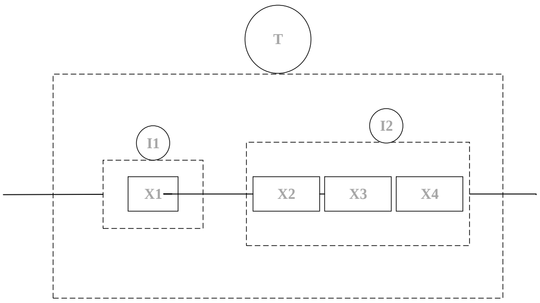

Fig .4. SRBD of analog to digital

System reliability is:

$$\mathrm{I1} = \mathrm{Rx1},\ \mathrm{I2} = \mathrm{Rx2} * \mathrm{Rx3} * \mathrm{Rx4},\ \mathrm{T} = \mathrm{RI1} * \mathrm{RI2} \quad (8)$$

The linguistic variables which are assigned for severity, occurrence and detection by experts should be converted to a fuzzy format. Also, the weights of the criteria should change to fuzzy format. The resulting fuzzy numbers act as data for entering the fuzzy VIKOR technique.

Experts used ten-scale linguistic variables for evaluating the risk factors and their relative importance significances. They chose one linguistic variable based on their experience and insight; then linguistic variables were converted into triangular fuzzy numbers; which are shown in Table 7.

Table 7. Relation between linguistic variables and triangular fuzzy numbers

Triangular fuzzy numbers	Linguistic variables	
(0.0,0.0,0.1)	NI	Nearly impossible
(0.0,0.1,0.2)	R	Remote
(0.1,0.2,0.3)	L	Low
(0.2,0.3,0.4)	RH	Relatively low
(0.3,0.4,0.5)	M	Moderate
(0.4,0.5,0.6)	MH	Moderate high
(0.5,0.6,0.7)	H	High
(0.6,0.7,0.8)	M	Major
(0.7,0.8,0.9)	VH	Very high
(0.8,0.9, 1)	EH	Extremely high

Occurrence, Severity and Detection are considered to have equal weights, as expert’s belief. In a meanwhile, according to the FMEA, final score of risks is equal to multiple of Occurrence, Severity and Detection. Infact, they have equal weights. Then Table 8 presents fuzzy weight of the criteria.

Table 8. Fuzzy Weight of Criteria

Criteria	Fuzzy weights
Occurrence	(0.33 ,0.33 ,0.33)
Severity	(0.33 ,0.33 ,0.33)
Detection	(0.33 ,0.33 ,0.33)

Based on the evaluations of five FMEA team members about the importance of aggregate fuzzy ratings of four risk Factors, the fuzzy decision matrix and the aggregated fuzzy decision matrix were constructed as shown respectively in Table 9.

According to the Table 7, linguistic terms converted and aggregated to the triangular fuzzy number and decision matrix was constructed as Table 10. By applying Eq. (2), the fuzzy best and the worst alternative based on three criteria calculated as Table 11.

Table 9. Assessment Information on Four Risk Factor in Three Criteria by Five Experts

Criterion															
Risk factor	Occurence	Severity	Detection	Occurence	Severity	Detection	Occurence	Severity	Detection	Occurence	Severity	Detection	Occurrence	Severity	Detection
	E1			**E2**			**E3**			**E4**			**E5**		
X1	H	H	H	VH	VH	MJ	MJ	VH	VH	EH	H	H	MJ	MJ	H
X2	H	H	RH	VH	M	MH	H	VH	M	H	H	M	MJ	MJ	H
X3	EH	EH	L	VH	MJ	RH	MJ	VH	MH	EH	EH	R	VH	VH	L
X4	VH	VH	M	MJ	MJ	RH	H	H	M	VH	VH	MH	VH	MJ	MH

Table 10. Aggregate triangular fuzzy decision matrix

Risk Factor	Occurrence			Severity			Detection		
X1	0.64	0.74	0.84	0.6	0.7	0.8	0.56	0.66	0.76
X2	0.56	0.66	0.76	0.52	0.62	0.72	0.34	0.44	0.54
X3	0.56	0.66	0.76	0.72	0.82	0.92	0.16	0.26	0.36
X4	0.64	0.74	0.84	0.62	0.72	0.82	0.32	0.42	0.52

Table 11. Fuzzy Best Value and Fuzzy Worst Value

	Occurrence			Severity			Detection		
f^*	0.56	0.66	0.76	0.52	0.62	0.72	0.16	0.26	0.36
f^o	0.64	0.74	0.84	0.72	0.82	0.92	0.56	0.66	0.76

By assuming V=0.66, $\tilde{Q}_i$ is calculated for all risk factors using the presented equation, and the results are shown in the table 13. The next step is to defuzzify the triangular fuzzy number of $\tilde{Q}_i$ into Crisp number. In the final phase, the alternatives areranked by Q ̃_i index, and the values are shown in table 13.

Table 12. Index $\tilde{S}_j$and $\tilde{R}_j$

rank	R-crisp	$\tilde{R}_j$			rank	S-Crisp	$\tilde{S}_j$			Risk Factor
3	0.228	0.148	0.217	0.33	3	0.46	0.355	0.375	0.768	X1
1	0.08	-0.108	0.099	0.23	4	0.519	-0.503	0.099	0.602	X2
2	0.16	0	0.165	0.33	1	0.111	-0.338	0.165	0.452	X3
2	0.16	0.13	0.09	0.33	2	0.256	0.193	0.087	0.66	X4

Table 13. Index$\tilde{Q}_i$ and Rank

Risk Factor	$\tilde{Q}_i$			$\tilde{Q}_i$-crisp
X1	-0.01958	0.24523	0.99	0.36522
X2	0.6586	0.1293	0.8284	0.4364
X3	-0.12811	0.097	0.8259	0.2229
X4	0.20983	0	0.9339	0.2859

Based on the results obtained, and the method presented in section 4.1(C1, C2), it is observed that the risk factor, each of the four basic system events, has the highest rank and ranked in terms of risk, and should be risk analysis.

For this purpose, risk analysis and corrective actions for risk reduction are expressed in Section 6.

6. Risk analysis of highly risk events

In Figure 5, a simple presentation of the procedure of a software that is applied to send data from the sensor and receive data by micro as well as the communication path between them have been presented. For its failure, as discussed, the four events of X1, X2, X3, and X4 are risky events. For further review of these events, we will analyze their risks for a better understanding and simplicity by examining Figure 5. In this system, there is a probability of failure in Sender, Receiver, or the communication link between them.

In order to improve the performance and promote the reliability of data paths, the data transfer paths have increased to two paths, and to prevent CCF occurrence, the method of sending and communicating from two different mechanisms has been used as a redundant [26]. In later sections, the cause of the failure of all the above mentioned items has been investigated.

To improve performance and promote reliability, data transfer paths have increased to two paths, and to prevent CCF, the method of sending and communicating from two different mechanisms is used as a redundant. In later sections, the cause of failure in all of the above mentioned items has been investigated.

Fig . 5. General working procedure of the software

6.1. Failure in communication paths

As stated above, communication paths are initially considered to prevent the possibility of failure and increase the two-path reliability. Systems that are used to reduce the error rate, when more than one of them is used in the system, are called redundant.

Redundant systems fail due to two types of failures: independent and dependent. One of the most important dependent failures in redundant systems is a common cause of failure. This type of failure leads to concomitant failure in components of the redundant system. In other words, the failure of more than two components in redundant system, which occurs over a short period of time, is the common cause of failure. There are a number of factors in redundant systems that multiply the failure among the components. This widespread failure causes simultaneous failure of system components and causes problems in the whole system during the mission period. Consideration of the common cause of failure begins from the design phase and should minimize the factors causing common failure, calculate the incidence rate of common failure cause, and include it in assessment of reliability.

6.2. Failure in Sender

Analog pressure sensor is the Sender in this system. Failure in this sensor, which actually results in generation of false pressure data, can occur for a number of reasons: 1) Sensor failure; 2) False reading of correct pressure data due to improper design; 3) Occurrence of noise.

The proposed actions to prevent the occurrence of failure are as follows:

1. Use of two independent sensors
2. The box design and location of sensors are such that the pressure inside the box is equal to that outside the box.
3. Reading data in large numbers per unit time and then averaging the read data to reduce the effect of noise.

6.3. Failure in Receiver

Micro acts as the Receiver in this system, the failure of which can be a function of failure in each of its internal components, including AD MUX and ADC port.

To solve this problem, it is recommended to use two independent processors in the system. for example, two Micro and FPGA processors.

7 Conclusion

Nowadays, the detection of failure is more difficult as the CPSs become more and more complex. The software with millions of lines of code play a key role in the failure or success of a system; therefore, the goal is a safety software design from the very beginning of the software development cycle.

In this paper, we present a method for generating SRBD in the requirements analysis phase by use-case diagram and use-case specifications. Then, using the VIKOR-based FMEA method, we further examined the risk of SRBD base events.

The fuzzy VIKOR method focuses on ranking and selecting from a set of alternatives in a fuzzy environment. Imprecision in multi-criteria decision making is modeled using fuzzy set theory to define criteria and the importance of criteria (weights). The triangular fuzzy numbers are used to handle imprecise numerical quantities. The VIKOR method is based on the aggregating fuzzy merit $\tilde{Q}$ that represents distance of an alternative to the ideal solution. The fuzzy operations and procedures for ranking fuzzy numbers are used in developing VIKOR algorithm.

In our future work, we examine the reliability by developing a software using UML and converting it into a State Method.

Acknowledgments

The authors would like to thank the Aerospace Research Institute project for providing the software architecture as well as the operational use-cases of DCU system.

Reference

[1] Rajkumar, R., Lee, I., Sha, L., & Stankovic, J. (2010, June). “Cyber-physical systems: the next computing revolution,” I*n Design Automation Conference (DAC), 2010 47th ACM/IEEE* (pp. 731-736). IEEE.

[2] Wu, F. J., Kao, Y. F., and Tseng, Y. C. (2011). “From wireless sensor networks towards cyber physical systems,” *Pervasive and Mobile computing*, 7(4), pp. 397-413.

[3] Murali D V. Verification of Cyber Physical Systems, Unpublished Master of Science Thesis, Virginia Polytechnic Institute and State University, Blacksburg, Virginia, 2013.

[4] Kim, H., Wong, W. E., Debroy, V., & Bae, D. (2010, November). “Bridging the gap between fault trees and UML state machine diagrams for safety analysis,” *In Software Engineering Conference (APSEC),* 2010 17th Asia Pacific (pp. 196-205). IEEE.

[5] Dowson, M. (1997). “The Ariane 5 software failure,” *ACM SIGSOFT Software Engineering Notes*, 22(2), 84.

[6] Anthony, M., Arno, R., Dowling, N., & Schuerger, R. (2012, May). “Reliability analysis for power to fire pump using fault tree and RBD,” *In Industrial & Commercial Power Systems Technical Conference (I&CPS)*, 2012 IEEE/IAS 48th (pp. 1-7). IEEE.

[7] Fazlollahtabar, H., & Niaki, S. T. A. (2018). “Fault Tree Analysis for Reliability Evaluation of an Advanced Complex Manufacturing System,” *Journal of Advanced Manufacturing Systems*, 17(01), 107-118.

[8] Yang, Z., Bonsall, S., & Wang, J. (2008). “Fuzzy rule-based Bayesian reasoning approach for prioritization of

failures in FMEA," *IEEE Transactions on Reliability*, 57(3), 517-528.

[9] Wang, Z., Gao, J. M., Wang, R. X., Chen, K., Gao, Z. Y., & Zheng, W. (2018). "Failure Mode and Effects Analysis by Using the House of Reliability-Based Rough VIKOR Approach," *IEEE Transactions on Reliability*, 67(1), 230-248.

[10] Deng, X., & Jiang, W. (2017). "Fuzzy risk evaluation in failure mode and effects analysis using a D numbers based multi-sensor information fusion method," *Sensors*, 17(9), 2086.

[11] Opricovic, S. (2011). "Fuzzy VIKOR with an application to water resources planning," *Expert Systems with Applications*, 38(10), 12983-12990.

[12] Tiwari, S., & Gupta, A. (2015). "A systematic literature review of use case specifications research," *Information and Software Technology*, 67, 128-158.

[13] Towhidnejad, M., Wallace, D. R., & Gallo Jr, A. M. (2003, December). "Validation of object oriented software design with fault tree analysis," *In null* (p. 209). IEEE.

[14] Vyas, P., & Mittal, R. K. (2012, March). "Eliciting additional safety requirements from use cases using SFTA," *In Recent Advances in Information Technology (RAIT), 2012 1st International Conference on* (pp. 163-169). IEEE.

[15] Kim, H., Wong, W. E., Debroy, V., & Bae, D. (2010, November). "Bridging the gap between fault trees and UML state machine diagrams for safety analysis," *In Software Engineering Conference (APSEC), 2010 17th Asia Pacific* (pp. 196-205). IEEE.

[16] Romani, M. A. D. S., Lahoz, C. H. N., & Yano, E. T. (2010). "Identifying dependability requirements for space software systems," *Journal of Aerospace Technology and Management*, 2(3), 287-300.

[17] Oveisi, S., & Ravanmehr, R. (2017). "SFTA-Based Approach for Safety/Reliability Analysis of Operational Use-Cases in Cyber-Physical Systems," *Journal of Computing and Information Science in Engineering*, 17(3), 031018.

[18] Liu, H. C., You, J. X., You, X. Y., & Shan, M. M. (2015). "A novel approach for failure mode and effects analysis using combination weighting and fuzzy VIKOR method," *Applied Soft Computing*, 28, 579-588.

[19] Safari, H., Faraji, Z., & Majidian, S. (2016). "Identifying and evaluating enterprise architecture risks using FMEA and fuzzy VIKOR," *Journal of Intelligent Manufacturing*, 27(2), 475-486.

[20] Rajput, B. S., & Chourey, V. (2015). "UML based Approach for System Reliability Assessment," *International Journal of Computer Applications*, 131(2).

[21] Liu, H., Deng, X., & Jiang, W. (2017). "Risk evaluation in failure mode and effects analysis using fuzzy measure and fuzzy integral," *Symmetry*, 9(8), 162.

[22] Ford Motor Company. "Potential Failure Mode and Effects Analysis (FMEA), Reference Manual; Ford Motor Compony: Dearborn, MI, USA, 1988.

[23] Kun, Z. H. A. N. G., Weiren, K. O. N. G., Peipei, L. I. U., Jiao, S. H. I., Yu, L. E. I., & Jie, Z. O. U. (2018). "Assessment and sequencing of air target threat based on intuitionistic fuzzy entropy and dynamic VIKOR," *Journal of Systems Engineering and Electronics*, 29(2), 305-310.

[24] Liao, H., Xu, Z., & Zeng, X. J. (2015). "Hesitant fuzzy linguistic VIKOR method and its application in qualitative multiple criteria decision making," *IEEE Transactions on Fuzzy Systems*, 23(5), 1343-1355.

[25] Wang, C. H., & Pang, C. T. (2011). "Using VIKOR Method for Evaluating Service Quality of Online Auction under Fuzzy Environment," *International Journal of Computer Science Engineering & Technology*, 1(6).

[26] Zhu, P., Han, J., Liu, L., & Lombardi, F. (2015). "A stochastic approach for the analysis of dynamic fault trees with spare gates under probabilistic common cause failures," *IEEE Transactions on Reliability*, 64(3), 878-892.

[27] Oveisi, S., & Ravanmehr, R. (2017). "Analysis of software safety and reliability methods in cyber physical systems," *International journal of critical infrastructures*, 13(1), 1-15.

[28] Kamandi, A., Azgomi, M. A., & Movaghar, A. (2006). "Transformation of UML models into analyzable OSAN models," *Electronic Notes in Theoretical Computer Science*, Vol. 159, 3-22.