Developing a Software Safety Analysis Model for Nuclear Imaging Device Based on Markov Chain

Gerami Farzad¹, Nikbakht Mohammad², Abedpour Elnaz³, Mordi Parisa³ and Moradi Faradonbeh Niloufar³

¹Department of Industrial Engineering, Naghshe Jahan Higher Education Institute, Esfahan, 8144167984, Iran.
²Planning and Development Department, Nasr Company, Isfahan, Iran, +98-3133246570.
³Student of Industrial Engineering Department, Naghshe Jahan Institute of Higher Education, Baharestan, Isfahan, Iran, +98-3136826055.

Received: 30 Jun 2016 Revised: 29 April 2016 Accepted: 19 May 2016

*Address for correspondence
Gerami Farzad
Department of Industrial Engineering, Naghshe Jahan Higher Education Institute, Esfahan, 8144167984, Iran.
Email: gerami@naghshejahan.ac.ir

This is an Open Access Journal / article distributed under the terms of the Creative Commons Attribution License (CC BY-NC-ND 3.0) which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. All rights reserved.

ABSTRACT

Using software in instruments and equipment in the world is growing rapidly and it can be said of the new generation of software development, safety and reliability applications, such as other engineering products is an important issue in many countries. Software use in a device must be safe and reliable and able to prevent unsafe conditions and high maintenance costs and even life events. For example, if you can imagine that a software error leading to an imaging device, that can lead to excessive radiation exposure to the patient and cause irreversible damage. According to the importance of the topic and also the few studies conducted in this area, in this article, a software safety analysis model based on Markov chains is developed. Then the model for nuclear imaging device in a state medical imaging center implemented and risk rates, probability, reliability, mean time to failure and the severity-repeated index have been calculated on the software immune system.

Key words: Softwaresafety, reliability, severity-repeated index, nuclear imaging device.

INTRODUCTION

In many systems and equipment, including medical equipment, correct function of the software is important because simple incorrect function may be because a large percentage of losses such as death, injury or environmental
damage. For example, heart patients due to improper software programs available in the pacemaker, have lost their lives. So we have to increase the reliability of the software. However, it should be noted that all software errors do not lead to safety problems and all software functions are not safe of what they are defined, but the goal of design, construction and implementation of the software safety program is eliminating hazards or reduce the risk to an acceptable level, and the safety engineering systems reducerisks by identifying, monitoring and evaluation, and reach the potential effects of adverse incidents to a minimum. Accidents that are associated with medical devices are classified into seven distinct groups as shown in Figure 1.

Safe area include necessities such as:
1. Reliability
2. Warn or prevent high risk output
3. Accuracy of measurement

This research aims to discuss software safety in nuclear medical devices. Nuclear imaging is one of the imaging methods in medicine that is a powerful and non-invasive tool that can give information to doctors for physician's diagnosis through metabolic and functional changes in defect organs. Imaging or nuclear scan is a technique that via safe and a very small amount of radiation provides detailed images of the body. It should be noted that the amount of radiation that patients receive through imaging is equal with common CT-scan and amount of ray that ordinarily receive through 2 or 3 years. Small amounts of radioactive material inter into the bloodstream are called radionuclides, radionuclides attached to red blood cells and with them circulate in the heart. A special gamma camera that is sensitive to the return ray of radionuclides, receives gamma rays. (The gamma camera detects and counts photons originating from the target organ and takes an image from single Scintillation created). Nuclear imaging device can be divided into different types, that in this study, Spect CT and PTC models are briefly mentioned Spect CT is a technology in nuclear medicine and radiology and the fusion of tomography and Spect. Spect CT systems take simultaneously functional and anatomical images from the patient. Spect CT acquires functional information from Spect and anatomical information from CT. Spect CT is composed of CT scanner, separate gamma camera and common board. The PTC is a device that enables to provide simultaneous imaging of anatomical lesions (such as location, size and shape) and the metabolic changes that occur mainly in cells. There is a number of software in the nuclear imaging device that control with main software. This study presents a mathematical model based on Markov chain to evaluate and improve the level of safety in the above software. Numerous studies have been conducted in the field of security software that some of them are mentioned below.

In Borcsok and Schaefer (2007) research has mentioned that reliability analysis of the software systems safety often needs to add expert information; because little information is valuable basis. In this study, information expert finds analysis process by using data. This extra information is able to calculate reliability characters with the most accurate count. [1] Adler and Kemmann (2009) explained that the software safety guarantee hope to approach the development of safety software. The software evaluation by the ability of its independence is a challenge. They could design Argument software through combine general software with an engineering model. [2] Paul et al (1999) in their paper discussed the analysis of the danger that includes safety aspects of the software, the reliability of the programs and data safety necessity and finally design a model and test confidence coefficient (tolerance) of the system. [3] Valentin et al (2008) conducted a study to examine the safety test in Petri Nets to access the system LHC. [4] Sun et al (2009) a method for enhancing the reliability of software on molding SRGM for safety critical software represented which help how to use the software reliability growth model that becomes NPP PSA. [5] Park et al (2009) believe that each section of the four additional networks that built base on DRPS have four main processors such as BP processor, CP processor, ATIP processor and the COM processor. Each operation is done by BP and CP and other processors have an indirect relationship with the operator. So any adverse action in BP (or CP) software, can change the result DRPS. They believe that decreasing error in the software development is so important because software error is due to design failure. [6] Menon et al (2009) in their study evaluate the safety standards and the results are discussed in the data obtained basis of providing evidence. [7] Habliand Kelly (2010) in their paper
suggested that warranty process can range from an operator by a clear definition of the software between perfect and without default software. This operator confirms the status change of perfect software until its warranty.[8]

Yongchao et al (2010) in their paper on the safety of the existing software reliability test had gained by technology (which is aeronautical) were examined.[9]

Medikonda et al (2010) in their study analyzed integrated safety, which critical system control software (which is safe FTA test) for determining the input data was checked.[10]PanesarWalawege et al (2010) were able to obtain the chain descriptions of existing evidence for safety software.[11]Ploesch et al (2010) were successful to provide a way to continue a code quality management by using safety analysis software.[12]Smith and Simpson (2011) were successful to provide a simple guide for safety software that emphasizes on standard automatic measurement tool to integrate examination for safety in the development of a software project.[13]Mayr (2011) in his paper safety standard operating software for the operating system is presented.[14]Denney and Pai (2011) described evaluation of the software safety because of having trustable Argument.[15]Knight et al (2011) in their study tried to invent a new way to improve the safety.[16]Jang and Park (2012) presented a method for estimating the reliability in nuclear software safety. This method is based on reliability software resulted of SRGM model. They investigated the capability of the following:

1) Mold design based on a special method to ensure highly accurate estimates
2) The ability to predict software errors [17].

Park et al (2013) evaluated probability failure of the Bayesian software. The purpose of this method is insuring the reliability of the software.[18]

Hawkins et al (2013) demonstrated that in the software safety, there is an assurance to each of the rules proportional to system risks distribution in software, which is dependent on preparing trustable Argument.[19]Hawkins et al (2013) developed a prototype of a software safety Argument for the aircraft wheel brake system.[20]Habli et al (2013) studied on the principles of the software safety assurance.[21]Thus, according to the above mentioned, need for software safety assessment in nuclear imaging device to protect the life and health of the patient, the safety of personnel and property of the medical center seems necessary.

Analysis of Safety Software

Software safety analysis is divided into three levels: system level, model level, functional level.

Analysis of the system level

"Risk" is a safety feature. The combination of the risk and the probability of its occurrence are index value of it. The system level usually includes fault tree analysis, failure mode, effects, and analysis of the crisis and event tree analysis based on risk.

Analysis of the model level

In the process of analyzing the model level, "risk" is the important safety feature. Acceptable level of risk is usually used to measure the system safety.

\[ \text{THR} = \frac{10^{-8}}{\text{hour} \times \text{system basic function}} \]
Analysis of the functional level

In this level “safe fail” is base of safety features. Using similar technology that is used to estimate the safety is the features of reliability. Most of the time assumes any risk of failure, is only two possible failure modes with different possibility:

Safe fail and unsafe fail. Dangerous failure rate (FRDS = λ) and relative risk (δ = Ʌ) and probability of safe keeping (PKS) and mean time to dangerous side failure (MTTF DS = ∫₀^∞ PKS(t) dt) are four quantitative indexes for measuring the system safety. Analysis of the functional level needs a clear definition of possible consequences of two different results of failure. Until the safe and unsafe boundaries are clearly defined, we can construct the confidence hypothesis by using probability theory. But risk is only a relative concept that this method can prevent analyzing of layer unit, thus we need to consider two factors, severity and frequency of our risk. As a result, there are advantages and disadvantages of these three levels of analysis. So over system safety analysis, system futures must be considered. Using this method, continuously with system analysis, we can select one of the three-levels or different combination of them.

Survey method (Development of safety analysis model)

In this study a safety analysis model base on Markov chains (transition from one stage to another) has been developed. Markov chain is a series of random variables that all random variables have the same sample space, but their probability distributions can be different. Each random variable in a Markov chain is only dependent on just prior variable. The sequence of random variables represents as follows:

\[ X^{(0)}, X^{(1)}, X^{(2)}, \ldots \]

The sample space of random variables of Markov chain can be continuous or discrete, limited or unlimited. Due to the nature of the problem under consideration, we will assume discrete and finite sample space. But the subject is also extended to the case of continuous and unlimited. Assuming discrete finite sample space, we can show any random variable with a probability distribution. We present this distribution with a vector (P) that includes the probability values of each sample space. The other display of the Markov chain is:

\[ P_0, P_1, P_2, \ldots \]

According to the definition of the Markov chain, knowing the first component of the chain and the relationship that makes component from component is enough. \[ P_1 = [p(X^1 = x_1) \ldots p(X^1 = x_n)] \]

This relation calls conversion function (T) and this is how to obtain the probability vector components by the following function:

\[ p(X^{i+1} = x) = \sum p(X^i = x) T_i(x, x) \]

Graph of different software systems

For an application under study considered three different modes that were shown in the figure below. This graph also mentions the use of Markov chain in safety analysis and software reliability.

Model assumptions

For the graph above, the following assumptions are considered:

- Imaging machine is composed of a software system and an immune system
- System (machine) fails when the software system fails to work.
- All fails are independent constantly.
- All failure rates are constant.

In the following table indexes and variables used in this model are presented.
Reliability

The reliability of a system is the possibility of safe and perfect operation for specified time according to the existing and predetermined conditions.

\[ R(t) = 1 - \sum_0^t f(t) \]

\( f(t) \) is equal to the probability density function that obtains from a Poisson distribution.

\[ f(t) = e^{-\lambda t} \frac{d^t}{dt} \]

\( \lambda = \) the number of failures occurred during the time \( t \).

Failure rate

To obtain a failure rate in relatively high equipment from during the performance, until the disability or failure to act of the machines a survey has done and equipment failure rate at time \( t \) is the probability density of failure in the next time interval, provided it is perfect at the beginning of the period.

\[ \lambda(t) = \frac{P(t)}{R(t)} \]

Repair rate

The mode of failure of the device at any given time determines the principles and methods.

\[ \mu(t) = \int_0^t e^{\lambda t} dt \]

In formula 0 is equal to the average length of service.

The mean time to failure of the system software

This is a good measure for estimating the average time that a software system working before the error in each of the states is defined in the graph above.

Safety analysis model equations

According to the different transitions of the Markov chain and a software system and also taking into consideration the variables and parameters are defined, the following calculation is performed using the technique of differential equations:

\[ \frac{dP_0(t)}{dt} + (\lambda_u + \lambda_s)P_0(t) = 0 \]

\[ \frac{dP_1(t)}{dt} - \lambda_s P_0(t) = 0 \]

\[ \frac{dP_2(t)}{dt} - \lambda_s P_0(t) = 0 \]

By solving the equations we get:

\[ P_0(t) = e^{-(\lambda_u + \lambda_s)t} \]

\[ P_1(t) = \frac{\lambda_s}{\lambda_u + \lambda_s} \left( 1 - e^{-(\lambda_u + \lambda_s)t} \right) \]

\[ P_2(t) = \frac{\lambda_u}{\lambda_u + \lambda_s} \left( 1 - e^{-(\lambda_u + \lambda_s)t} \right) \]

Similarly, the reliability of the software system is:

\[ R(t) = e^{-(\lambda_u + \lambda_s)t} \]

\( R(t) \) is the reliability of the software system at time \( t \).

The mean time to failure of a software system is:
MTTF<sub>xm</sub> = \int_0^\infty R(t) \, dt = \frac{1}{\lambda_x + \lambda_y}

MTTF<sub>xm</sub> the mean time to failure of the system software. The table below shows the types of failures, time and duration of the service provided. Based on the above table and equations obtained, we can present the calculations in the table below. The results suggest the probability that a software system works normally is equal to 0.924, the probability that a software system fails safely is equal to 0.063 and also the possibility that software system fails unsafely is equal to 0.0129.

Calculation of FSI index for modes 1 and 2

In relation to the failures above mentioned, indexes AFR, ASR and FSI in modes 1 and 2 calculates as below (over 3 years): In the calculation of the FSI index for the failures mentioned in modes 1 and 2 of the diagram of the transmission system of software system that is the appropriate index in comparative studies, we can say that the failures in which the software immune system does not work can play important role in estimating safety related to the software. Although the number of failures that the immune system does not work is less than that immune system work and fewer lost hours are allocated to them, but their FSI index almost half. Based on Gerami and Rocky (2010), if the FSI index in a year is less than 0.1 indicates the safety level is relatively high. Consequently, according to calculations have been done for 3 years, it can be said imaging device under consideration is not desirable for safety level.

CONCLUSION

Nuclear imaging devices because of the wide workplace, unpredictable movements and the nature of the control computer program of them have certain characteristics. Thus, the existing software on nuclear imaging devices causes unique challenges in terms of safety. In this study based on Markov chain a software safety analysis model for the nuclear imaging device was developed. This model can be used in conjunction with other safety analysis techniques, to enhance the security of the software. Also for developed model, based on data sampled at a government medical imaging center, hazard rates, possibilities, confidence capabilities, mean time to failure, and also intensity-repeated indices related to software immune system were calculated and results presented in tables 3 and 4. Based on these results, strongly recommend that imaging center managers immediately take action to develop effective programs to enhance the software safety level. For future research suggests, the effectiveness of the model in comparison with other safety analysis techniques can be calculated and this model can be used for other devices and sensitive equipment such as control software of fighter aircraft or surgical robot.

ACKNOWLEDGMENTS

We would like to express our sincere appreciation for the following people: Mrs. Masumeh Abedini and the other personnel for their valuable help for providing us data for this study.

REFERENCE

Those causes of accidents in medical equipment

Figure 1 - Causes of accidents in medical equipment

Figure 2 - Software system transition from one stage to the next
Table 1- indexes and variables

<table>
<thead>
<tr>
<th>Description</th>
<th>Variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>$i$ is the $i$th state of the system</td>
<td>$i$</td>
</tr>
<tr>
<td>$i = 0$: Software system works normally</td>
<td></td>
</tr>
<tr>
<td>$i = 1$: Software system has failed safely</td>
<td></td>
</tr>
<tr>
<td>$i = 2$: Software system has failed unsafely</td>
<td></td>
</tr>
<tr>
<td>$P_i(t)$: The possibility that the system is in state $i$ at time $t$</td>
<td>$P_i(t)$</td>
</tr>
<tr>
<td>$\lambda_i$: The $i$-th constant failure rate</td>
<td>$\lambda_i$</td>
</tr>
</tbody>
</table>

Table 2 - Types of failures, time and duration of the service

<table>
<thead>
<tr>
<th>Row</th>
<th>Fail / Potential risk</th>
<th>Occurrence time</th>
<th>Length of service</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>The existence of magnetic waves</td>
<td>Every four months</td>
<td>30 minutes</td>
</tr>
<tr>
<td>2</td>
<td>Login incorrect data</td>
<td>Every six months</td>
<td>1 day</td>
</tr>
<tr>
<td>3</td>
<td>System board failure</td>
<td>Every four months</td>
<td>8 days</td>
</tr>
<tr>
<td>4</td>
<td>Operator error</td>
<td>Rarely Once a year</td>
<td>1 day</td>
</tr>
<tr>
<td>5</td>
<td>Data redundancy</td>
<td>Rarely Once a year</td>
<td>2 days</td>
</tr>
</tbody>
</table>

Table 3. Calculated values

<table>
<thead>
<tr>
<th>Definition</th>
<th>The amount</th>
<th>Record</th>
</tr>
</thead>
<tbody>
<tr>
<td>Danger rate from 0 to 1</td>
<td>0.52</td>
<td>$\lambda_s$</td>
</tr>
<tr>
<td>Danger rate from 0 to 2</td>
<td>0.107</td>
<td>$\lambda_u$</td>
</tr>
<tr>
<td>System reliability</td>
<td>0.925</td>
<td>$R(t)$</td>
</tr>
<tr>
<td>The mean of fail time</td>
<td>1,594</td>
<td>$MTTF_{xm}$</td>
</tr>
<tr>
<td>Possibility to work normally</td>
<td>0.924</td>
<td>$P_0$</td>
</tr>
<tr>
<td>Possibility of safe failure</td>
<td>0.063</td>
<td>$P_1$</td>
</tr>
<tr>
<td>Possibility of unsafe failure</td>
<td>0.0129</td>
<td>$P_2$</td>
</tr>
</tbody>
</table>

Table 4. Calculation of intensity-repeat index for modes 1 and 2 (FSI)

<table>
<thead>
<tr>
<th>State of</th>
<th>Number of events</th>
<th>Lost work days</th>
<th>Lost work time</th>
<th>Useful work time</th>
<th>AFR</th>
<th>ASR</th>
<th>FSI</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>13</td>
<td>93</td>
<td>12875</td>
<td>116700</td>
<td>111.396</td>
<td>48.50</td>
<td>2.324</td>
</tr>
<tr>
<td>2</td>
<td>8</td>
<td>55</td>
<td>7140</td>
<td>112500</td>
<td>71.11</td>
<td>23.75</td>
<td>0.884</td>
</tr>
</tbody>
</table>