Chapter 3

Reliability and Safety: State-of-the-art and Perspectives

In the past several decades, significant attention has been devoted to the quality assessment of Safety-Critical and Control Systems (SCCS) from many perspectives such as its reliability, safety, and performance. The systematic review work conducted in this thesis has been carried out for asserting state-of-the-art and perspectives in the field of reliability and safety for Computer Based System (CBS) under consideration of the thesis. Researchers are continuing to put their efforts to ensure these dependability attributes. This Chapter summarizes the state-of-the-art in the field of reliability and safety of such systems. It also discusses the feasibility study for the applicability of existing models on safety-critical systems. A detailed literature survey is conducted to investigate the various techniques/models to ensure the reliability as well as safety of the CBS. The limitations of these models are also analyzed with respect to their applicability in safety-critical systems, for which a

case study of Nuclear Power Plant system has been taken. The direction for future research is also suggested that is based on the case study to extend the further scope of research. This chapter provides concepts and understandings that are required for working out analysis and propositions in the rest of this thesis work.

3.1 Introduction

A Safety-Critical and Control System (SCCS) executes critical tasks, whose failure could endanger human life, lead to substantial economic loss, or cause extensive environmental damage. Table 1.1 of Chapter 1, lists notable catastrophic accidents that have taken place over the last several decades. Today, these systems are digital systems and are being used in the field of military, nuclear power plants, medical, etc. The dependability of a system is a measure of its ability to commence and complete a mission without failure. It can be thought of as the quality of the system that permits the user to rely on it for service. In a case of safety-critical systems, reliability, safety, and performance requirements are very high.

The reliability of the system is often defined as the probability that the system does not fail in a given environment, during a specified exposure time interval [1]. Whereas, the safety of a system is the conditional probability that the system has survived the interval during an exposure time interval without an accident, given that it was operating without catastrophic failure at start time [2].

Reliability or safety breaches may lead to catastrophic disaster. The impact of reliability or safety breaches varies from minor inconvenience (e.g., no warm water for a bath) to potential disaster, viz. personal injury, substantial economic loss, mission failure, and death. Reliability or safety breaches that have received potential disaster include [12], [13], [14]: Shutdown of the Hartsfield-Jackson Atlanta International Airport, Loss of Communication between the FAA Air Traffic Control Center, and Airplanes, Loss of the Mars Polar Lander, Loss of the Mars Climate Orbiter, Misplacement of a Satellite by Titan IV B-32/Centaur Launch Vehicle, and many more. The list of such failures due to reliability or safety breaches are endless. Table 1.1 of Chapter 1, lists notable catastrophic accidents which demonstrate that reliability or safety breaches are one of the most important causes to such mishaps. This table also includes severity impact, significant losses and probable reasons by which these mishaps occur. In most of the disaster cases, investigating team have come to conclusion that these mishaps could have avoided by the field data availability, reliability and safety analysis, incorporated with prompt functionality, or mitigate the severity level due to reliability or safety breaches. There are numbers of reliability or safety failures cases which don't make the news and don't have the significant disaster, but even though it may cause: substantial customer's inconvenience, large warranty cost expenditure by a company in terms of monetary and human resources, impact on the good will of an organization, etc.

Performance is defined as the total effectiveness of a computer system, including throughput, individual response time, and availability [43]. Measuring reliability,

safety and performability can be used for planning and control all testing resources during system design and development. Therefore, such systems must be highly dependable.

Since the 1960s, several models have been proposed for reliability analysis of the systems. Whereas, researchers started to propose models for safety from the early seventies. Each model is meant for specific kind of system and contains certain limitations. No model is generic that fits in all kind of systems.

3.2 Reliability and Safety Concepts:

Reliability depends on failure rate, operating time and environmental conditions in which product is operational [44]. Reliability is the probability of a component/subsystem/system performing its required function under the stated operating environment for a specified duration of time [45]. The reliability can be quantified in terms of mean time to failure (MTTF).

Reliability has been an identified performance variable for at least 50 years. Von Braun [46], [47] proposed probably very first reliability model for complicated Vengeance Weapon 2 (V-2) missile system on the failure of its first version known as the Buzz-Bomb (V-1) missile. Later on, Pieruschka [47] changed the model and proved that under specific presumptions, the reliability of a system is equal to the product of the reliabilities of its elements. It was the first recorded modern predictive

reliability model. Thereafter, other researchers expanded the work and developed numerous reliability models. As the dependency and complexity of SCCS increase the demand for reliable subsystems and used parts increased. Many studies have been carried out, and even facets of a mathematical concepts are discovered for disintegrating the system into subsystems [48].

Failure rate plays a vital role in formulating the mathematical models for reliability analysis. Various standards governing by state government and Industry sectors reliability and safety domains that have received great exposure include: 1)MIL-STD-721 [44] which specifies failure rate as the ratio of the overall number of fails within a product population and the overall number of life units (complete operating time) used by that population, throughout a specific measurement period under specified conditions; 2)IEC 61709 [44] standard is all about usages of failure rate data and stress models for conversions for reliability prediction of electronic components. Whereas, this standard fails to provide specific failure rate estimation; 3)NASA-STD-8719.13C [49] safety standard gives us guidance for software acquisition and development of safety critical systems. It provides necessary data, various software activities and detailed documentation; 4)MIL-STD-882E [50] safety standard practice provides the different approaches for identification of hazards, elimination of possible hazards by Department of Defense, USA. The standard covers broad areas of possible hazards as they apply to: system, equipment, infrastructure throughout the system development life cycle, use and its disposal. Table 3.1 [44], [51], [52] lists various industrial and government regulatory bodies standards used

TABLE 3.1: Various standards used in reliability prediction methods based on application and applied industry

Method	**Applied Industry**	**Specific intended application**
MIL-Hdbk-217	Military	Reliability prediction of electronic equipment
Telcordia SR-332	Telecom	Reliability prediction procedure (RPP) for electronic equipment TR-332
China 229 B	Military	Reliability data for components used in telecommunication systems
CNET	Ground Military	Reliability table for semiconductor devices
SAE Reliability Prediction Method	Automotive	By use of field return data, it predicts reliability for automotive electronic products
BT-HRD-5	Telecom	Uses for comparing the potential reliability of electronic equipment
Siemens SN29500	Siemens products	Reliability and quality specifications failure rates of components
PRISM Commercial	Military	System reliability assessment methodology
HIRAP Commercial	Aviation	Reduces uncertainty in reliability prediction of hardware used

in reliability prediction based on application and applied industries.

Safety is the internal property of a system however a safe system can't be guaranteed. In any case, if in some system risk of damage to life, environment or property may be controlled and brought inside of as far as possible limits, then this sort of system can be called as safe. Nowadays, safety is an essential concern in automation industries as a result of ability to execute normally as well as abnormally, without risk of causing human injury or death and without harm to external/system environment [53]. NPR 8715.3C and MIL-STD-882D [53], [54] define - safety is freedom from those conditions that can cause death, injury, occupational illness, damage to or

loss of equipment or property, or harm to the environment. During earlier days, issues related to safety were termed as "reliability engineering" or "dependability engineering". Terrey [55] introduced conventional safety engineering. Leveson [56] stated that system safety engineering methods could be useful when utilize this to software safety because proven approaches have already been repeatedly taken to work with on software. However, in a practical application safety could merely quantify by the considering system as whole, i.e. both software as well as all hardware parts. The study of the SCS is not mature as comparing of research issue distinguished from system safety and computer reliability. Knight and Littlewood [57] discuss most of issues related to the SCS. Levensons [58] provides a comprehensive bibliography and it is most often cited article. Bown and Stavride [59] gives a summary of the industrial utilisation of formal techniques and significant no. of standard engineering in the safety critical area. Safety requirements are exclusive requirements i.e. they exclude unwanted scenario rather than particular required system services. Safety-critical systems are those systems, where safety concerns are more important than the functionality of the system. While safety considerations and reliability consideration may result in overlapping requirements, they could also lead to different or even contradictory requirements. The essential safety requirement, independence from accidents, is qualitatively not the same as the reliability requirement concerning continuity of the necessary service.

Reliability and safety are distinct system concepts: the former describes how well the system performs its function and the latter states the system functions usually

do not lead to an accident. A system could be reliable but unsafe. An illustration of this kind of system is an NPP system, which continues to function under adverse conditions such as core cooling systems failure, however, directs an increase in temperature and pressure to run the NPP in spite of core cooling systems failure. The system itself might be reliable; it is functioning, however, leads to an accident. The system will be considered safe (in this case) if, on detecting the core cooling systems failure course, a new course was computed to the removal of heat from the fuel to mitigate the melting of the core. Likewise, a system might be safe but unreliable. As an example, a railroad signaling system may be entirely unreliable but safe if it always fails in the most restrictive manner; in other words, whenever it fails it shows "stop". In this instance, the system is safe even though it is not reliable.

Safety; unlike reliability, is a system property, not a component property [60]. Investigating the NPP is safe within acceptable limit, for e.g., it is not possible by examining only a spray nozzle of a coolant sub-system in the NPP. But, the reliability of a spray nozzle is meant. However, safety can be evaluated using a relationship between the spray nozzle and reaming components of the NPP, in the context of the system.

The main objectives of reliability or safety prediction [15], [16], [17], [44], [45], [48], [61] are as follows:

- Find out the feasibility of meeting the reliability or safety requirements.

- Determine whether a specific design meets the target reliability or safety requirements.
- Used to compare different topologies, control strategy and components.
- Help to manage the system operation and maintenance.
- Used to predict warranty cost and maintenance support requirements.
- Assessment of potential risks.
- Provide input to safety analysis.
- Establish reference for logistic support requirement (e.g., maintenance, components, and upgrades).

Today, several methods are available but no method guarantees to fulfill all the objectives.

3.3 A Case Study

In this section, we give a complete case study of DFWCS as safety critical control system along with its failure modes. All the possible failures occur due to failure of intended function of hardware, software, and or both. We use this case study to check whether the considered existing reliability prediction methods are able to predict effective reliability of the system.

3.3.1 DFWCS Overview

A Pressurized-Water Reactor (PWR) uses a Digital Feed-water Control System (DFWCS). The primary function of DFWCS is to regulate the flow of feed-water during normal at power operations, and optionally during plant heat up or cool down and the schematic diagram is shown in Figure 3.1. The DFWCS uses the heat generated in the reactor core to derive the turbine for the power generation that serves two steam generators (SGs). Every SG has its specific digital feedwater controller. The digital feedwater controller is required to keep up the water level inside the SG and guarantee that it is inside the marked water level range. The controller is considered to be failed if the water level in the SG is outside the marked range. The digital feedwater controllers are associated with: 1) feedwater pump controller (FP), 2) main feedwater valve controller (MFV), and 3) bypass feedwater valve controller (BFV). The controller deals with the stream flow of feedwater to the SG to keep up a steady water level. The FP pumps the feedwater through the high-pressure feedwater heaters into the SGs. The MFV and BFV manage the amount of feedwater flowing off to the SG to keep up a consistent water level in the SG.

3.3.2 Operating Modes

Depending on the generation of power in primary system, the DFWCS operates in two different modes on the basis of operational point of view. The two automatic

modes of operation are 1)low power automatic mode (2–15%) power - operating in three-element (SG level, feedwater flow, and steam flow) controls, and 2)high power automatic mode (above 15%) power - single-element (SG level) control. In low power automatic mode, MFV is closed, BFV is opened, and the FP is regulated at a minimum speed. Whereas, MFV is opened and BFV is closed in the case of high power automatic mode.

3.3.3 Physical Connections for the DFWCS

To provide information about the various component's state activities of the controlled process to DFWCS, different types of sensors used. Figure 3.2 shows these components connection with the computer. These sensors measure feedwater level, steam flow, FP status and feedwater temperature. The computer gives input signs to the FP, BFV, and MFV through an analog control signal and failure status signals. The PC is associated with a Watchdog (hardware) timer is used to stop the process during hardware or software failure.

3.3.4 Possible Failures for the DFWCS

Water level of the feedwater tank is continuously sensing by level sensor and send information to the computer to take appropriate actions, open or close the valve for water flow in the tank. During this process there are 10 possible failure events due to failure of software, hardware, and or both. These events are: 1) *E1:* MFV_fail

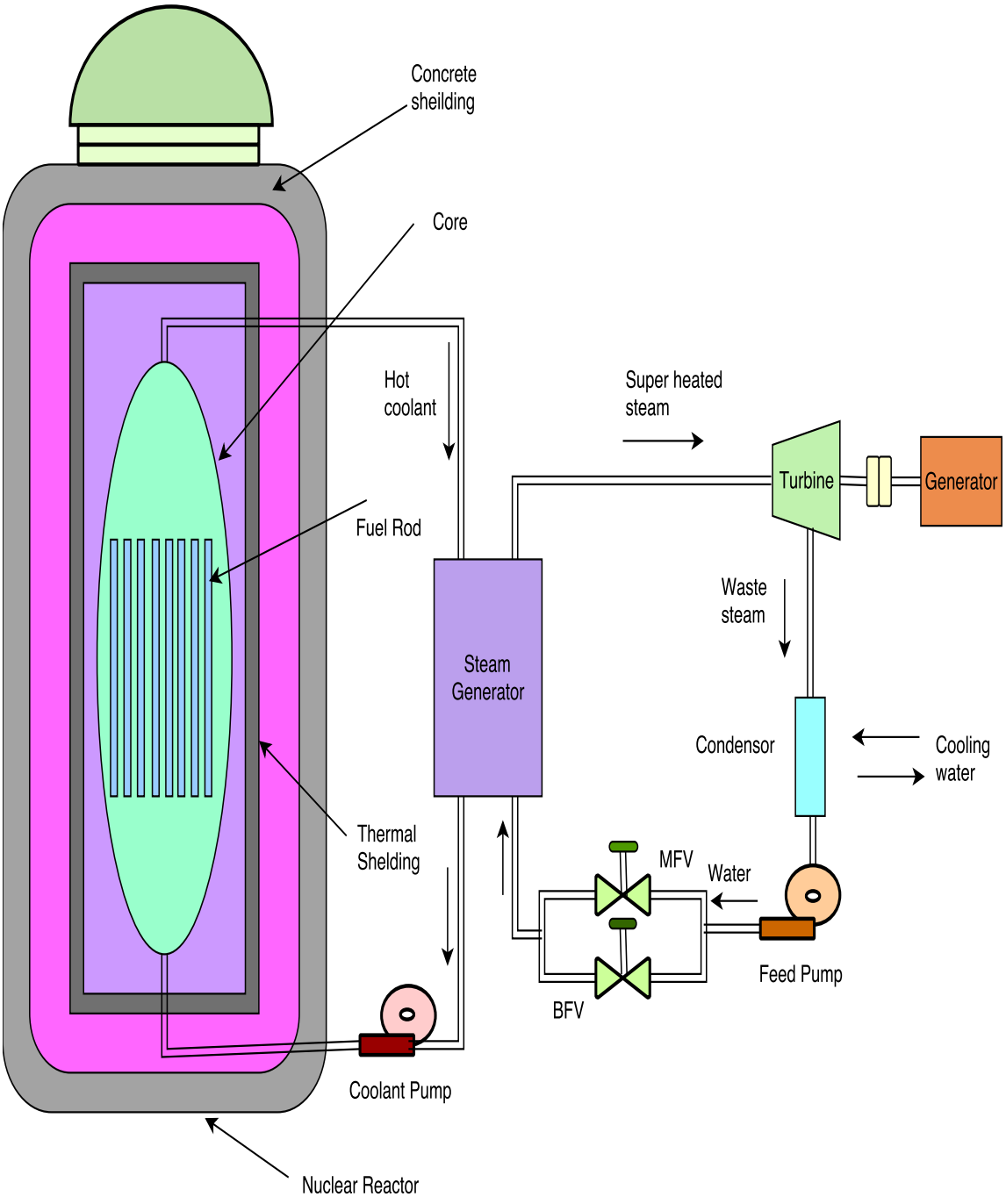

FIGURE 3.1: The DFWCS system outlay

to close, 2) *E2:* MFV_ fail to open, 3) *E3:* FP_fail to off, 4) *E4:* FP_fail to on, 5) *E5:* BFV_fail to close, 6) *E6:* BFV_fail to open, 7) *E7:* Computer_fail to working, 8) *E8:* Sensor_fail to sense, 9) *E9:* Water below the minimum marked level, and 10)

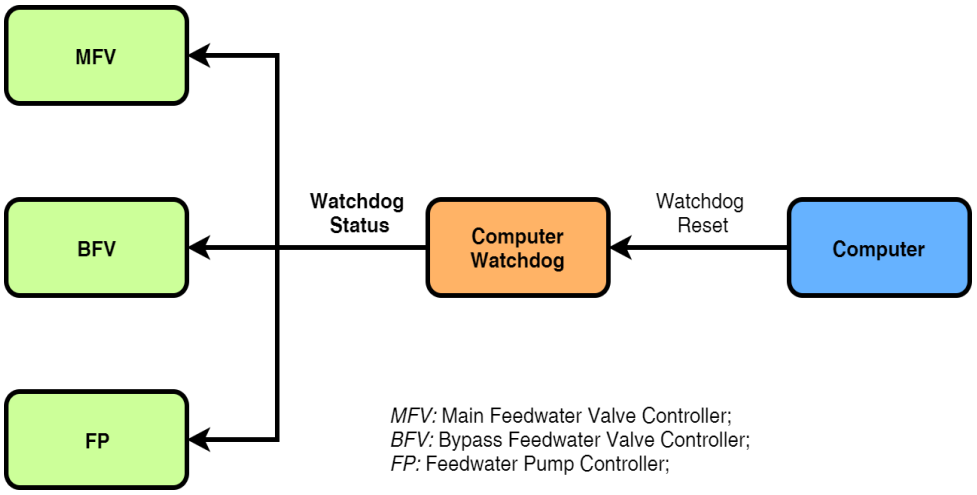

FIGURE 3.2: Digital feedwater controller status interconnections for computer

E10: Water reach the maximum marked level.

3.4 Reliability Prediction: State-of-the-art and Perspectives

In this section, we give a brief overview of existing reliability prediction approaches along with their merits (based on case study discussed in concerned approach) and limitations (based on applicability of DFWCS as case study). The methods to overcome their limitations, with respect to their applicability on DFWCS, are provided. Software failures are supported by some of techniques and some techniques supported for hardware failures. A few of existing techniques are supported for system that composed of both hardware as well as software. Broadly the exiting techniques for reliability prediction consists of 4 phases as shown in Figure 3.3, that we also followed. We have explained each phase with respect to our case study.

3.4.1 Phase 1: Requirement analysis of the SCCS

In this phase, we gather all the requirements of a SCCS which is to be build. We try to capture all the high level requirements into unambiguous, consistent, complete, traceable, and stake-holders approved requirements.

3.4.2 Phase 2: Partitioning the SCCS into software components, interfaces, and hardware components

In this phase, we break our SCCS in to software and hardware components/-subsystems along with various interfaces. Thereafter the appropriate techniques can be applied simultaneously to perform reliability estimation of software and hardware components along with interfaces individually.

3.4.3 Phase 2.A: Software system analysis

In this phase, our goal is to predict reliability for software systems (actually subsystems) only. For this, we attempt to develop specific model, which can be best suited.

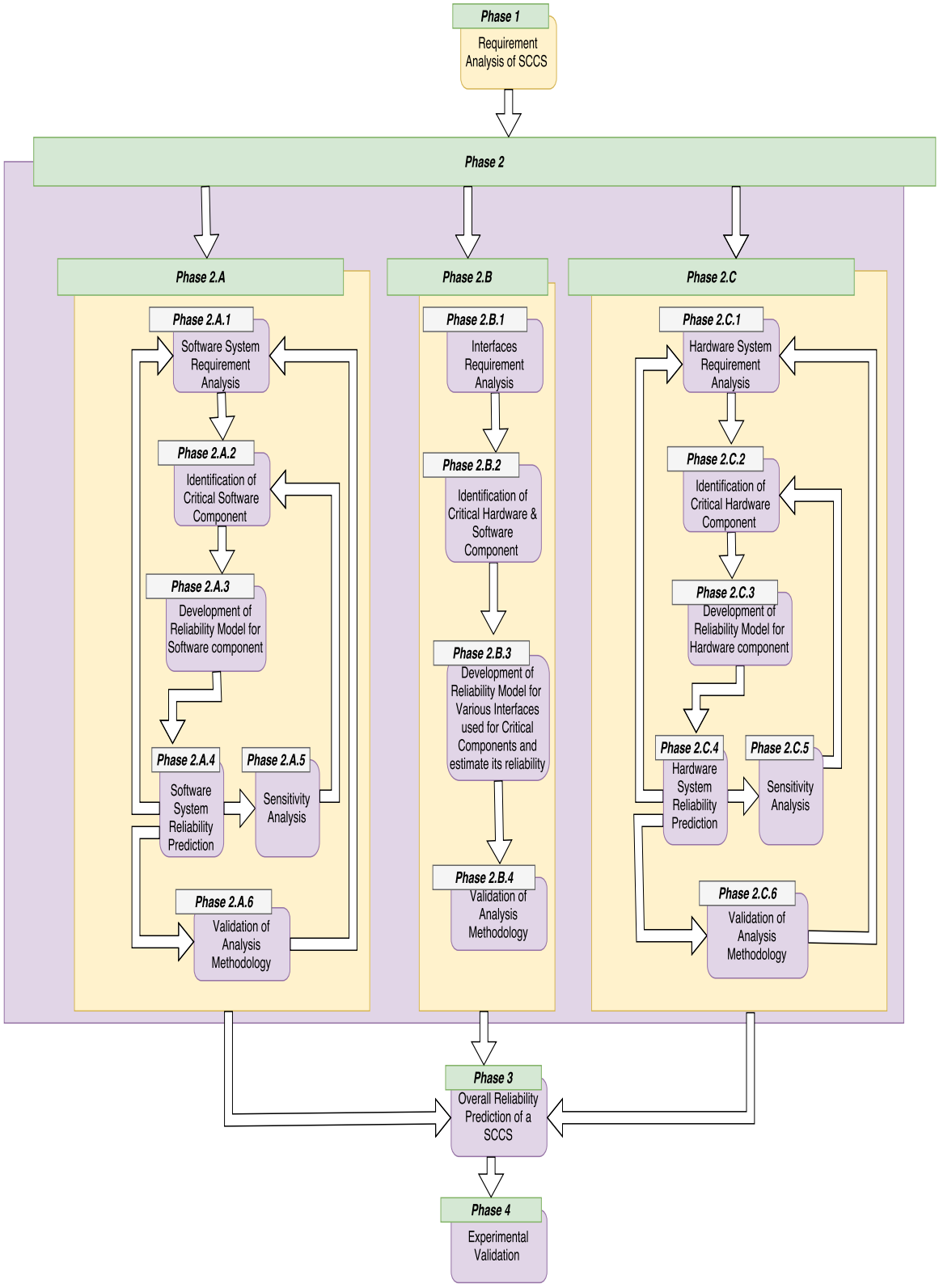

FIGURE 3.3: Reliability Prediction Framework for SCCS

3.4.4 Phase 2.A.1: Software system requirement analysis

In this phase, we gather all the information about software components that are required to support functionalities of SCCS. All the high level requirements of the software system are analyzed. This is the first step for the software reliability estimation approach.

3.4.5 Phase 2.A.2: Identification of critical software components

In this phase, we try to identify the critical components of software, the vulnerability of which may leads to SCCS failure and hence can result into high possibility of catastrophic disaster. Generally such identification process have been done by ranking or various modeling techniques. This information will help the design phase of software development life cycle to outline these critical components precisely, by consolidating redundancy, and so on.

3.4.6 Phase 2.A.3: Development of reliability model for software system

In this phase, we developed the software reliability model. This software model uses various proposed algorithms and different reliability models that incorporate

our suggestions to overcome the limitations of existing approaches for the accurate software reliability estimation of SCCS.

3.4.7 Phase 2.A.4: Software system reliability prediction

In this phase, we predict the software reliability from the model that is developed in the previous phase. The transition probability matrix from the state space hybrid model is used to find overall software reliability. If the estimated reliability is not up to the target reliability then re-visit the phase 2.A.1 and take appropriate actions such as design modification, component(s) replacement, etcetera to improve reliability of the system.

3.4.8 Phase 2.A.5: Sensitivity analysis

In this phase, we perform sensitivity analysis to know the impact of a change in software component(s) reliabilities on the software reliability of the overall system. This helps to identify critical components which might missed in earlier phases. If any critical component is found missing, we re-iterate from phase 2.A.2.

3.4.9 Phase 2.A.6: Validation of analysis methodology

In this phase, structural and behavioral properties of the developed software reliability model based on requirements can be analyzed. Analysis of state space

model gives valuable knowledge, viz. – deadlock freedom, liveness, boundedness, and etcetera. It is also used to verify the various important properties such as mutual exclusion, etc. In case of missing any functional requirement (s), we re-iterate from phase 2.A.1. Various important quality attributes that are essential for SCCS like safety, reliability, security are quantitatively estimated.

3.4.10 Phase 2.B and its sub phases: Reliability prediction for interfaces

In this phase, we formulated a reliability model for all the interfaces.

3.4.11 Phase 2.C and its sub phases: Reliability prediction for hardware system

In these phases, we formulated a reliability model for all the hardware components.

3.4.12 Phase 3: Overall reliability prediction for SCCS

In this phase, we gathered the estimated reliability of all the three subsystems: 1) software components, 2) interfaces, and 3) hardware components and estimate the overall reliability of the SCCS.

3.4.13 Phase 4: Experimental Validation

In this last phase of proposed framework, we validate the accuracy of the estimated result and correctness of our developed approach. We use the operational profile of 2 years for DFWCS to compute its reliability.

Table 3.2 gives a comparative study of existing reliability analysis approaches based on their system type. In spite numerous papers were dedicated to reliability analysis, most of them either do not include numerical demonstrations, or demonstrate the models on non-real examples. A few papers of them take the real case studies and perform experimental validation with the theoretical results. Table 3.2 also provides the details of the validation of existing techniques. Table 3.3 discusses the pros and cons of the existing techniques along with the suggestions to overcome the limitations.

The summary of identified approaches for reliability prediction is as follows.

Clifford [62] proposed a mathematical modeling to predict the failure rate of component parts. Earlier mathematical modeling has been developed at two levels: 1) system- collection of components, assemblies and/or subsystems arranged in a particular design in order to get required functions; or black box level- used for analysis purpose without knowing its internal structure of assemblies, subassemblies, and/or components; 2) effect of physics of failure at atomic and nuclear levels of stresses on materials. In this paper, author proposed a new intermediate level

related to probable failure rates of component parts to their reliability character. Proposed mathematical model for evaluating part failure rate is:

$$\lambda_{\mathrm{p}} = \mathrm{C.Q.}\left(\left[\mathrm{e}^{\left(\frac{\mathrm{T}}{\mathrm{n_T}}\right)^{\mathrm{D}}}\right]\left[\mathrm{e}^{\left(\frac{\mathrm{S_1}}{\mathrm{n_T}}\right)^{\mathrm{E}}\left(\frac{\mathrm{T_0}}{\mathrm{T_1}}\right)^{\mathrm{EF}}}\right]\left[\mathrm{e}^{\left(\frac{\mathrm{S_1}}{\mathrm{n_1}}\right)^{\mathrm{E}}\left(\frac{\mathrm{T_0}}{\mathrm{T_1}}\right)^{\mathrm{EF}}}\right]\left[\mathrm{e}^{\left(\frac{\mathrm{S_2}}{\mathrm{n_2}}\right)^{\mathrm{G}}\left(\frac{\mathrm{T_0}}{\mathrm{T_1}}\right)^{\mathrm{GH}}}\right]\left[\mathrm{e}^{\left(\frac{\mathrm{S_3}}{\mathrm{n_3}}\right)^{\mathrm{J}}\left(\frac{\mathrm{T_0}}{\mathrm{T_1}}\right)^{\mathrm{JK}}}\right]\right).\lambda_{\mathrm{b}} \quad (3.1)$$

where,

C : a constant for particular part design;

Q : a quality adjustment factor for a particular supplier;

n_i: knee value of specific stress response;

T : temperature of body in kelvin;

D : temperature degradation;

S_ii: Non thermal stress factor;

E, G, J : non thermal stress interaction failure;

F, H, K : temperature stress interaction factor;

T_0/T_1: ratio of operating temperature to normal degrading temperature in kelvin.

***Merits*:**

This approach is applied to all types of system components. In this case, attributes and constants that are used in failure rate prediction are derived using empirical test.

Using this model, accurate failure rates for any specific application may be predicted. In addition, the fast prediction is possible with actual stress on components.

***Limitations*:**

There are two limitations associated with this model when we are going to use for SCS which are: (i) to develop such model requires strong mathematical background and hence is susceptible to use in the industrial especially in complex systems like that of DFWCS and (ii) operational failure data of system components is required which is most of time very difficult to get due to confidentiality of DFWCS. (iii) As in DFWCS, the system contains heterogeneous components that include hardware, software and firmware. The hardware failure rate is constant during operational time which is not applicable for softwares and firmwares and hence it will not give the effective prediction.

***Suggestions*:**

To apply this method on DFWCS, the operational profile data of similar systems may be used for operational failure data. If there is insufficiency of data of existing system, extrapolation techniques [63] may be used. Monte Carlo Simulation [64] may be used to handle the mathematical complexity of the model.

M. Faraji et al. [65] presented an infrastructure performance oriented reliability assessment method using weighed stochastic Petri net (WSPN) model. It is a new type of coordinated modeling approach for simulating the reliability of vital infrastructure spatially lifelines for a hazard and even the succeeding interdependencies

amongst the interconnected infrastructures. In various other comparable sorts of strategies, the weight of network elements is not specified or sometimes is appointed by professional suggestion or complex network analysis. In this model by utilizing the basic graph theory parameters, weight of each element is specified in stochastic Petri nets. As a result, the cascading effects throughout the network as well as reliability can be evaluated based upon weighed stochastic Petri nets. They used mathematical model and transition probability for SPN analysis. If state S_j is directly t_k reachable from state S_i then transition probability is given by:

$$q\left(S_i\ ,\ S_j\right) = \frac{r\left(t_k\right) * n_i\left(t_k\right)}{\sum_{t \in T} r\left(t\right) * n_i\left(t\right)} \tag{3.2}$$

where,

T: set of transitions;

r: firing rate function that provide firing rate r(t) to every transition t;

n: firing rank function (no. of active firing for each transition);

***Merits*:**

This model has an ability to represent the relationships dynamically between infrastructure elements. Due to the coupling of multiple interdependent infrastructure elements, it can address infrastructure protection, mitigation, response, and recovery issues. Using this model, we can assess cascading impacts throughout the network

and reliability prediction. We can use this technique to address cascading failure effect on reliability prediction on the component as well as on the whole system.

***Limitations*:**

The limitation with this model is to obtain Life/failure data of the components, as in case of DFWCS. That is because safety critical systems confront very less number of failures due to its robust design to meet high reliability requirements. Also, since the system contains COTS component, its failure rate is difficult to obtain. Identifying common mode failure requires much effort in terms of domain knowledge of how components of DFWCS fail?

***Suggestions*:**

The life data of component of the system like DFWCS can be predicted by the modified Coffin-Manson method [66] and the modified Ostegren method [67]. Further, more accurate life data prediction can be achieved by using Artificial Neural Network (ANN) approach utilizing both failure and suspension condition monitoring histories [68].

Andre Kleyner et al. [69] created a model to forecast reliability of occupant safety system with partial detection and repair. Traditional approaches of approximating system reliability do not consider the impact of fault detection ability & punish the inclusion of detection circuitry because of the higher parts count. In this paper, authors determined system availability, which could be connected to the system's possibility of failing on demand P_{fd}, which is a better choice to predict the reliability.

They used dynamic scenario with aging time consideration with the assumptions that system is under warranty and all the required repair will be performed with renewal attrition function:

$$(t) = \begin{cases} 1.0; \text{ where, } t < T_w \\ f(t); \text{where, } T_w < t < T_{life} \end{cases} \tag{3.3}$$

where,

t: time in year of services;

T_w: warranty term duration;

T_{life}: expected vehicle life;

f(t): repair function; $0 \leq f(t) \leq 1$ and $f(Tw) = 1$;

After passing warranty, repair population reduce by (t),

$$P_{fd}^{pd} = \theta\rho(t)\ P_{fd}^{s}(t) + (1-\theta\rho(t))[1-R^{nr}(t)] \tag{3.4}$$

where,

$P_{fd}^{pd}(t)$: probability of failure on demand when all the failures are detected;

P_{fd}^{s}: probability of failure on demand for system, which does not go under repair;

θ: fraction of system will go for repair;

R^{nr} (t): reliability of system under no repair condition;

***Merits*:**

The proposed technique combines numerous real-life aspects, such as probability of the occupant to observe the caution signal, reliability of detection circuitry, occupant's reaction time to the caution light, period of repair service, approximated down time, system age, and various other pertinent factors. This illustrates the level of sensitivity of chance of failure on demand to different elements. This model offers a more practical and versatile approach to estimate the system's failure rates and hence reliability, as compared to the more conventional reliability evaluation techniques. SPN gives a visual traceability of the solution as compared to some stochastic approaches, such as customized Monte Carlo simulation.

***Limitations*:**

Authors claim that field data is not required. However, attrition function cannot be developed without analysis of warehouse shipping history of product. Many SCSs are confidential in nature, therefore, applicability of this approach on DFWCS reliability prediction is very difficult due to confidentiality of field data. Further, authors used constant failure rate of the components, which is a non-conservative approach and causes less accuracy. DFWCS contains heterogeneous components that include software, hardware and firmware. Therefore it is not suitable for such applications.

***Suggestions*:**

If failure data of DFWCS is not available, it can be predicted analytically using a combination of: system specifications, COTS components [70] used elsewhere, similar type of running system with associating uncertainty [71], and non-failure constant rate of component .

G. Ramos et al. [72] proposed an approach for security evaluation of EIS and power systems based on Petri nets. The methodology not only suggests to model the operating series of protection devices, but also to model the unpredictability in the operation of protection devices using General Stochastic Petri Net (GSPN). It also proposes a technique to determine its influence on the security analysis. The proposed technique also permits the security evaluation of industrial electrical systems and power systems, taking into account of hidden failures and the sequence of operation in protection devices.

***Merits*:**

Earlier, reliability strategies model disruptions in a probabilistic means; however, they do not model the stochastic response of the power system [73] but this method does. The system is examined under steady-state conditions after the disruptions take place [74], [75], [76]. It was not feasible to specify indicators that consist the temporal response of the EIS when unexpected disruptions take place, which has been addressed in this technique.

***Limitations*:**

Forced stopping the safety critical systems is not affordable to many applications like nuclear power plant, aerospace, etc. DFWCS shut down is very costly (in terms of monetary, human life, etcetera). So, system should not react to each & every failure. This limits the practical applicability of this method for such systems.

***Suggestions*:**

There should be a decision mechanism to decide to shut down the DFWCS based on the severity. If the cause is not severe, it shall continue operation in a degraded fashion and during the course of time, it should undergo recovery process. The mechanism may consider the cascading fault effect of components [65] to other components to predict the consequences of the fault for severity analysis. Further, redundancy [77] can be incorporated to most critical components to avoid frequent shutdown. We may use partial fault detection technique [69]. If we use WGSPN et.al. [65] instead of GSPN modeling we can find out the criticality of the component that fails.

Bing Wang et al. [78] developed reliability model for electric vehicle motor by using fault tree and Extended Stochastic Petri Net (ESPN). Earlier, the research on reliability modeling of electrical automobile motor was restricted to FT evaluation method. However, they have come out with the following limitations of FT analysis.

- In FT analysis, the probabilities of fundamental events need to be known prior to analysis. Due to this assumption we are unable to get the real-time description of reliability information [79], [80].

- It is challenging for FT analysis to carry out additional quantitative evaluations because of an absence of effective methods of mathematical expression.
- FT analysis fails to model the dynamic faults of the system and specifically is incapable to explain the propagation process of faults.

These limitations are addressed in this paper. Let there are n components in a system, having life $x_1, x_2, x_3, \ldots\ldots\ldots, x_n$. A model is proposed in which healthy state of the system depends on x_i, $i \in 1, \ldots, n$. Then life of the system for AND / OR transitions in ESPN is evaluated from Fault Tree as:

For AND transition: $x = \min(x_1, x_2, x_3, \ldots\ldots\ldots, x_n)$ & for OR transition: $x = \max(x_1, x_2, x_3, \ldots\ldots\ldots, x_n)$. FT is converted into ESPN to predict the reliability.

***Merits*:**

It overcomes the limitations of FT. Less number of elements are required to build ESPN model as compared to FT generation. Association of life distribution to the transitions provide real-time capabilities that is a common requirement of SCCS.

***Limitations*:**

The authors did not validate this method using actual reliability test data. Another drawback of this method is reliability assessment of an integrated system that contains hardware and software components. Further, in the proposed method FT is converted into ESPN and therefore there is a possibility to miss the other

dynamic scenarios, which are not possible to model in FT. Also, it is not suitable for repairable systems as FT is incapable to model the repairable systems. Hence, it is not applicable to SCCS like DFWCS.

***Suggestions*:**

The proposed methodology can be modified to overcome the foresaid limitations with the help of Petri nets [81]. Petri net is a powerful tool to model the dynamic faults and repair. Several tools are available to do the stationary analysis of the Petri net models.

R. Kumar et al. [82] used Markov Analysis and failure characteristics of wear out components with Weibull distribution to get an accurate and effective technique for system reliability modeling. Generally, earlier reliability models were based on constant failure rates in which the probability of a part failure remains independent of the history of the component operation. In this approach, a mathematical model is proposed in which non-constant hazard rate is used. A probability density function f (t) under Weibull distribution can be calculated as:

$$\mathrm{f}\,(\mathrm{t}) = \mathrm{a.b.t}^{(\mathrm{b}-1)}\ .\exp\left(-\mathrm{a.t}^{\mathrm{b}}\right); \quad t > 0 (3.5)$$

where,

a: scale parameter;

b: shape parameter;

t: time to failure;

Hazard rate h(t) can be calculated as [83], [84]:

$$\mathrm{h}\,(\mathrm{t}) = \ \mathrm{a.b.t}^{(\mathrm{b}-1)}; \qquad t > 0 \tag{3.6}$$

From hazard rate, Markov model can be solved for non-constant hazard rate.

***Merits*:**

The non-constant failure rate of the component is considered. The proposed mathematical model takes relatively insignificant time as compared to Monte Carlo simulation method (greater than 1000 trials to achieve accuracy to 5th or 6th decimal).

***Limitations*:**

The authors used Markov chain, which is difficult to use in complex systems such as DFWCS due to the state space explosion problem. Therefore, it is very hard to manage the states of a large system (e.g. 6 components causes 64 states of Optical Telescope Calibration System (OTCS) system). Further, Markov chain follows exponential distribution which may not be practically applicable to distributed systems, which is true for DFWCS also. Therefore, this methodology does not provide more confidence on accuracy of the reliability prediction of DFWCS.

***Suggestions*:**

SPN method [85] may be used to overcome the limitations of Markov model. It also increases the modeling power to model several features like concurrency, multithreading, etc. which are essentially required for the modeling of DFWCS.

Zengkai Liu et al. [86] proposed DSPN model to evaluate the performance of subsea Blown-out Prevention system. In this paper authors break the system into two subsystems: mechanical system and computer based system, to obtain the availability and reliability of the system. Impact analysis of component failure rate and repair time on the overall system performance is also analyzed. The statistical mean number of components that fail in the shock is given by:

$$\mathrm{M_j} = \sum_{\mathrm{i=1}}^{\mathrm{k}} \mathrm{i} * \mathrm{P_j}\,(\mathrm{n},\ \mathrm{i}) \tag{3.7}$$

where,

$\mathrm{M_j}$: stastical mean no. of component that under shock;

j : denotes the component;

i : no. of components that's fail in shock (k n);

$\mathrm{P_j}\ (\mathrm{n}, \mathrm{i})$: probability of i component failure with n component in shock;

System shock rate,

$$\mathrm{V_j} = \mathrm{n.}\ \frac{\mathrm{j}}{\mathrm{M_j}} \tag{3.8}$$

where,

$_j$: failure rate of single component.

Failure rate of i component when n component in shock:

$$\lambda(n,i) = V_j * P_j\ (n,i) \tag{3.9}$$

***Merits*:**

In this paper, authors consider human error for reliability analysis of a repairable system to overcome the limitations of earlier approaches [87]. Further, earlier approaches use FTA technique [88], [89] which requires real failure data. This is not applicable here. FMEA technique fails to differentiate a situation of common failure or severe failure caused by compound failures [90], whereas DSPN does.

***Limitations*:**

The authors assume that failure rate of the component is constant that is again not applicable in the case of DFWCS. However, the failure rate is not constant for a software components of DFWCS, as discussed earlier. Reliability of only subsystems are computed, which is not sufficient to derive the overall system reliability of DFWCS. The reliability of interfaces must also be computed for evaluating the overall system reliability.

Suggestions

To get high prediction accuracy for DFWCS, Weibull distribution [91] should be used instead of using constant failure rate. The interfaces should be clearly defined with respect to their reliability estimation [92]. Then the reliability can be estimated based on the arrangements of the components or subsystems.

A. Mihalche et al. [93] proposed a model for mechatronic system to estimate the reliability using stochastic Petri nets. In Petri net models, the failure times are required to estimate the reliability of a mechatronic system. The method is validated on a case study: Antilock Brake System. This mechatronic system is broken into mechanical, electrical and computer system. The suggested technique is capable to evaluate the reliability for mechatronic systems. They used exponential distribution for the computing the reliability of electronic and software components whereas, Weibull distribution for the mechanical and hydraulic components.

The maximum likelihood estimators of the failure intensity, $\widehat{\lambda}$ is given by

$$\widehat{\lambda}\,(\mathrm{k}) = \frac{\sum_{\mathrm{i}=1}^{\mathrm{n}} \mathrm{D}\,(\mathrm{i})^{\mathrm{k}}}{\sum_{\mathrm{i}=1}^{\mathrm{n}} \sum_{\mathrm{j}=1}^{\mathrm{D(i)^k}} \mathrm{t}_{\mathrm{i,j}}^{(\mathrm{k})}} \tag{3.10}$$

where,

$\mathrm{D(i)}^{\mathrm{k}}$: the no. of failure for the i^{th} system at each failure of component k;

$\mathrm{t}_{\mathrm{i,j}}^{(\mathrm{k})}$: recorded times (k: component system index, i: system index, j: failure index);

Jelinski – Moronda model is used for software component, where the estimators $\widehat{\mathrm{N}_0}$ and $\widehat{\varphi}$ are calculated using maximum likelihood principle as:

$$\sum_{j=1}^{\max D(i)^{(k)}} \frac{\sum_{i=1}^{n} U\left(D(i)^{(k)} - j\right)}{\widehat{N_0}^{(k)} - j + 1} = \widehat{\varphi}^{(k)} \sum_{i=1}^{n} \sum_{j=1}^{D(i)^k} t_{i,j}^{(k)} \tag{3.11}$$

And

$$\widehat{\varphi}^{(k)} = \frac{\sum_{i=1}^{n} D(i)^k}{\left(\widehat{N_0}^{(k)} + 1\right) \sum_{i=1}^{n} \sum_{j=1}^{D(i)^k} t_{i,j}^{(k)} - \sum_{i=1}^{n} \sum_{j=1}^{D(i)^k+1} j t_{i,j}^{(k)}} \tag{3.12}$$

where,

U(x): the unit step function (equal to 0 $\forall$ x $<$ 0 and equal to 1 $\forall$ x$\geq$0);

N_0: Initial no. of faults;

φ: proportionality coefficient;

When time to failure is described by Weibull distribution, the estimators are calculated using maximum likelihood principle, as:

$$\frac{\sum_{i=1}^{n} D(i)^k}{\widehat{\beta}^{(k)}} + \sum_{i=1}^{n} \sum_{j=1}^{D(i)^k} \ln(t_{i,j}^{(k)}) = \left(\sum_{i=1}^{n} D(i)^k\right) \frac{\sum_{i=1}^{n} \sum_{j=1}^{D(i)^k+1} t_{i,j}^{(k)^{(k)}} \ln(t_{i,j}^{(k)})}{\sum_{i=1}^{n} \sum_{j=1}^{D(i)^k+1} t_{i,j}^{(k)\widehat{\beta}^{(k)}}} \tag{3.13}$$

And

$$\widehat{\eta}^{(k)} = \left(\frac{1}{\sum_{i=1}^{n} D(i)^k} \sum_{i=1}^{n} \sum_{j=1}^{D(i)^k+1} t_{i,j}^{(k)^{(k)}}\right)^{\frac{1}{\widehat{\beta}^{(k)}}} \tag{3.14}$$

where,

β: shape parameter;

η: scale parameter;

***Merits*:**

The proposed model is applicable for the whole system that contains mechanical, electronic, and software components. Use of estimated parameters, instead of the theoretical parameters adds another merit of validation, to this model.

***Limitations*:**

One of the limitations of this model is use of Jelinski – Moranda model that gives less accuracy in reliability prediction because it considers that the individual faults of the system are independent, which is not applicable in stochastic processes such as DFWCS. This model assumes that failed component is repaired immediately, which is also not practically feasible as some components takes even a week of time to repair or at least 5-6 hours for replacement. Therefore, it is not applicable for practical applications of DFWCS.

***Suggestions*:**

We suggest that each component should require a detailed analysis to obtain an accurate prediction result. Littlewood's Bayesian Differential Debugging Model [94] or Ramamoorthy and Bastani model [95] are the proven practical approaches, especially for safety critical systems like DFWCS.

K. Krishna et al. [96] proposed an approach for reliability early prediction of an application, the development of which is based on the output result of the prototype development using Rational Unified Process (RUP). Using this approach, the reliability is found to be significantly increased in the incremental cycles. In this paper quantitative software reliability pre-estimation is done using GSPN, based on RUP implemented prototype acquired from the PoC (proof of concept) of an economic application, before the real implementation of the application development.

***Merits*:**

This model uses output results of prototype development using RUP for the evaluation of the reliability of the application development, which significantly improves the accuracy of estimation.

***Limitations*:**

The prototype building requires a significant effort and resources for implementation of large sized systems like DFWCS. In this paper, authors consider a constant failure rate & constant repair rate, which is not valid in most of the safety critical systems that are composed of heterogeneous components like DFWCS and hence the accuracy of the prediction is questionable. Validation is not given to gain enough confidence to make it in practical use in the case of SCCS. The accuracy of reliability prediction is improved incrementally which is not acceptable to risk-based applications like DFWCS.

***Suggestions*:**

The model should accommodate the limitations of constant failure and repair rate to make it suitable for practical applications of SCCS like DFWCS. The approach should be demonstrated for the systems that contain heterogeneous components, which is more likely in case of industrial and safety-related applications. A mathematical validation is highly appreciated so that to gain confidence to implement such model for SCCS.

Chin-Yu Huan et al. [97] examined the unification of SRGM based on Nonhomogeneous Poisson processes NHPP. After that, he demonstrates how some SRGM, such as Goel-Okumoto model and inflected S-shaped model, could be obtained by using the principle of three widely prominent methods: weighted arithmetic mean, weighted geometric mean, and weighted harmonic mean. The fault detection sensation in the functional stage is different from that in the testing stage. In practice, software functional reliability is the primary issue for users [98], [99], [100], and [101]. Hence, the author further proposed a technique to define the transitions from the testing stage to the operational stage. This method could offer a valuable information that permits us to recognize the software failure behavior throughout its operational phase and provide a measurable analysis of failure distribution in the operational field. He modified other popular models as follows:

Goel – Okumoto model with multiple change points:

$$m\,(t) = a(1 - [-(b_n(t - \tau_{n-1}) + \sum_{i=1}^{n-1} b_k(\tau_k - \tau_{k-1}))]) \tag{3.15}$$

where,

$m(t)$: expected number of faults detected per test run;

a: expected no. of faults to be detected;

b_i : fault detection rate;

$(\tau_k - \tau_{k-1})$: observation time interval;

n: no of failures per sub interval;

And inflection s- shaped model with multiple change points:

$$\begin{aligned} \mathrm{m\,(t)} = a \Bigg(1 - & \frac{\psi * \exp\left[-\mathrm{b_n\ t}\right] + \exp\left[-\mathrm{b_n\,(t - \tau_n)}\right]}{1 + \psi * \exp\left[-\mathrm{b_n\ t}\right]} \\ & * \frac{\psi * \exp\left[-\mathrm{b_k\ \tau_k}\right] + \psi * \exp\left[-\mathrm{b_k\ (\tau_k - \tau_{k-1})}\right]}{1 + \psi * \exp\left[-\mathrm{b_k\ \tau_k}\right]} \end{aligned} \tag{3.16}$$

where,

ψ: inflection factor;

r: inflation rate

***Merits*:**

This model describes the transitions from testing phase to operational phase. It provides significant information that gives an idea to visualize the behavior

of software failure during operation and provide failure distribution in the field operation for quantitative analysis.

***Limitations*:**

One of the limitations of this approach is that it is unable to model the influence of the system usage profile on the control and data flow explicitly. The impact of a system's execution environment on reliability is not analyzed, which is essentially required in case of SCCS for accurate reliability prediction. Hence, not suitable for Safety- critical systems like DFWCS.

***Suggestions*:**

We can find parameter dependencies for usage profile by using Stochastic Regular Expressions (SRE) [102] as a modelling notation which can address the influence of the system usage profile on the control and data flow. Hence, accuracy of prediction will higher and applicability of such approach for SCCS like DFWCS is justified.

Wende Kong et al. [103] proposed an approach for early software reliability prediction using cause-effect graphing (CEEGA). This method is very helpful to predict software reliability having minimum information about the system at the end of requirement stage. In this paper, authors attempt to address two following realistic limitations of CEGA [104] from being widely used in the area of software reliability prediction:

1. There was no specific approach for identifying defects.

2. There was no quantitative method to link this measurement to software reliability.

In this model, the following two tasks are performed to quantify system failure probability:

1. Determining failure-relevant inputs and
2. The occurrence probability of all failure-relevant inputs.

He calculated probability of system failure as:

$$\begin{aligned}\Pr(\text{system fails}) &= Pr(\text{system fails}|\ \text{failure}-\text{relevant inputs}) \\ &\times Pr(\text{failure}-\text{relevant inputs}) + \Pr(\text{system fails}\,|\,\text{failure}-\text{irrelevant inputs}) \\ &\times \Pr(\text{failure}-\text{irrelevant inputs}) \\ &= 1 \times Pr(\text{failure}-\text{relevant inputs}) + 0 \times \Pr(\text{failure}-\text{irrelevant inputs}) \\ &= \Pr(\text{failure}-\text{relevant inputs})\end{aligned} \tag{3.17}$$

***Merits*:**

At requirement stage, prediction of software reliability has significant positive impact on cost, time & hard work. The proposed method can systematically identify defects in a Software Requirements Specification document. Another advantage

of this approach is that it "helps in recognizing requirements that are incomplete and ambiguous" [105]. Compared to the general SRS examination strategies such as ad hoc and checklist reading methods [106], cause-effect graph technique gives a more organized and clearer path for assessors to follow.

***Limitations*:**

Constructing an ACEG for a bulky SRS is very time-consuming. It appears an obstacle to figure out failure-relevant inputs manually for a dissimilar effect-pair when pertinent causes are greater than 15. The quantification is done at the very first stage of SDLC and hence only based on software requirements. Therefore, the accuracy of the prediction is less and hence it can be used for defense-in-depth purpose but not for SCCS like DFWCS, the failure of which may cause heavy losses.

***Suggestions*:**

First order predicate logic can be used for simplification, which can reduce time to some extent. The similar model may be proposed during the design phase as the level of abstraction goes down, to bring out the insight details of safety-critical system.

K. Saravan et al. [107] proposed a method, which has an ability to identify the errors that can hamper the reliability analysis. In this phase, a corrective action is taken at each phase of the SDLC, starting from requirements phase to coding phase. At the end of the coding phase, the faults are predicted. Operational profile is collected either from various sources, like testing phase, operational phase and from

the similar running system. The combination of operational profile and faults that are predicted at the end of the coding phase is used to predict the system reliability.

***Merits*:**

At the end of each phase, starting from the requirements to coding phase, there is an opportunity to take corrective actions. This will achieve the improved reliability estimate. This method also reduces the propagation of errors to proceeding phases.

***Limitations*:**

In this method, more precaution needs to be taken at the requirements phase, else the error propagates to the subsequent phases. In this paper, fuzzy profiles and rules are used, which give less accurate estimates. Hence, this method may prove more beneficial to non-critical applications rather than safety-critical applications like DFWCS.

***Suggestions*:**

Cause effect graph analysis technique [103] may be used at the requirements phase to ensure the correctness of requirements of SCCS. The assumptions should be made with proper justification and should be practical enough for specific kind of safety-critical application. However, validation of the method is essentially required to build confidence in the case of SCCS like DFWCS.

Yu Liu and Chu-Jie Chenet [108] proposed a model for non-repairable multistate system (MMS) to estimate the reliability using Bayesian method and aggregation

of inspection data dynamically. This proposed dynamic reliability estimation is two folded: 1) assessing the dynamic reliability of an individual MSS by concurrently or non-synchronous collection of inspection data aggregation that are extracted from multiple level of a system, and 2) the intended approach considers and take care of imperfections of inspection data at various physical levels of a system. This approach uses agglomeration of inspection data collected at multilevel for the dynamic reliability assessment and a two stage Bayesian method is used in recursive manner for the finding out the probable states of different components. The method is validated on Underground Flow Transmission System as a case study. Here, the aggregation of current observed indicator together with last observed indicator we can able to estimate the probability of a system at time t_k for a specific component's state combination. The conditional probability associated with this approach is given as below:

$$P\ [XS\ (t_k)\ = S_{i,\ v}\ |X^0(t_k)] \tag{3.18}$$

where,

$X_s(t_k)$: represent the states of components at i inspection time t_k;

$X^0(t_k)$: latest observed indicator at inspection time t_k;

$v \in \{1,\ 2, 3,\ \mathrm{L_i}\}$ **:** inspected system likelihood, holding the state i with the vth combination of components' states over given time period up to t_k.

$S_{i,\ v}$: states of components for specific combination of the components;

The following three cases would be possible for estimating the conditional probability of inspected system on the basis of last observed indicator:

Case 1: At any time (t_k) inspection performed at system level, the associated conditional probability of inspected system turns as follows:

$$P\ [X_s\ (t_k)\ = S_{i,\ v}\ |X^0(t_k)] = \ P\ [X_s\ (t_k) = \ S_{i,\ v}\ |X^0(t_k) = \theta_k^s,\ X^0(t_{k-1})] \quad (3.19)$$

where,

$X_s(t_k)$: represent the states of components at inspection time t_k;

$X^0(t_k)$: latest observed indicator at inspection time t_k;

$X^0(t_{k-1})$: Previous observed indicator at inspection time t_{k-1};

$S_{i,\ v}$: states of components for specific combination of the components;

θ_k^s : system level observed indicator at inspection time t_k;

Next, a Bay's method used to estimate the probability of system holding a state over the time t_k at system level as below:

$$\pi_{S,\ i}^k = \ \frac{\sum_{j=1}^{N_s} \pi_{S,j}^{k-1} p_{j,1}(\Delta t_k) b_{i,\ \theta_k^s}}{\sum_{j=1}^{N_s} \sum_{n=1}^{N_s} \pi_{S,j}^{k-1} p_{j,\ n}(\Delta t_k) b_{n,\ \theta_k^s}} \quad (3.20)$$

where,

$\pi^{k}_{S,\ i}$: likelihood of stay of the system at specific state i at time t_k;

$\pi^{k-1}_{S,j}$: likelihood of stay of the system at specific state j at time t_{k-1} ;

$p_{j,\ n}(\Delta t_k)$: system's probability for the state changes from state j to i in the time period $\Delta t_k = \ (t_k - t_{k-1})$;

$b_{n,\ \theta^s_k}$: quantifies imperfection data for system's state combination of n^{th} component at time t_k;

In such a way, we can update each elements at system level inspection from π^{k-1}_S to π^k_S at time $t_k$by use of Equation 3.20, defined in our case study.

Case 2: At any time (t_k) inspection performed at component level, the associated conditional probability of inspected system turns as follows:

$$\begin{aligned} P\ [X_s\ (t_k)\ &= S_{i,\ v}\ |X^0(t_k)] = \ P\ [X_s\ (t_k) \\ &= \ S_{i,\ v}\ |X^0(t_k) = \theta^l_k,\ X^0(t_{k-1})] \end{aligned} \tag{3.21}$$

where,

$X_s(t_k)$: represent the states of components at i at inspection time t_k;

$X^0(t_k)$: latest observed indicator at inspection time t_k;

$X^0(t_{k-1})$: Previous observed indicator at inspection time t_{k-1};

$S_{i,\ v}$: state of components for specific combination of the components;

θ_k^l : component level observed indicator at inspection time t_k;

Next, a Bay's method is used to estimate the probability of component l holding a state over the time t_k at component level as below:

$$\pi_{l,\ i}^{k} = \frac{\sum_{j=1}^{n_l} \pi_{l,j}^{k-1} p_{j,\ i}^{l}(\Delta t_k)\ \ a_{i\ ,\ \theta_k^l}^{l}}{\sum_{j=1}^{n_l}\sum_{m=1}^{n_l} \pi_{l,j}^{k-1} p_{j,\ m}^{l}(\Delta t_k)\ \ a_{m\ ,\ \theta_k^l}^{l}},\ i = 1,\ 2, \ldots, n_l. \tag{3.22}$$

where,

$\pi_{l,\ i}^{k}$: likelihood of component l at time t_k to stay at specific state i;

$\pi_{l,j}^{k-1}$: likelihood of component l at time t_{k-1} to stay at specific state j;

$p_{j,\ i}^{l}(\Delta t_k)$: component's probability for the state changes from state j to i in the time period $\Delta t_k = (t_k - t_{k-1})$;

$a_{m\ ,\ \theta_k^l}^{l}$: quantifies imperfection data for component's state combination of m^{th} component at time t_k;

Case 3: At any time (t_k) inspection performed at component, system, and/or both level, the associated conditional probability of inspected system turns as follows:

$X_s(t_k)$: represent the states of components, system, and/or both at i inspection time t_k;

$X^0(t_k)$: latest observed indicator at inspection timet_k;

$X^0(t_{k-1})$: Previous observed indicator at inspection timet_{k-1};

$S_{i,\ v}$: states of components, system, and/or both for specific combination of the components;

θ_k^s : system level observed indicator at inspection timet_k;

θ_k^l : component level observed indicator at inspection timet_k;

Merits:

This approach uses agglomeration of inspection data collected at multilevel for the dynamic reliability assessment. Since, it is impossible to observe all the actual state of a system (and/or components, assemblies, and subsystems) because of deformity of inspections. Hence, multilevel inspection data collection and using it leads to more accuracy for the reliability assessment. Due to use of a two stage Bayesian method in recursive manner for the finding out the probable states of different components and uses this inspection data to update the reliability function of a system dynamically gives more accurate estimated results in this approach.

Limitations:

The authors assume that system is non repairable, which is not applicable to most of the safety critical systems. Repair of the DFWCS takes places on failure due to its potential consequences, as discussed earlier. Another limitation is that this approach

does not consider the stagnation pattern of a system which sometimes might triggers by maintenance activities. This method gives less accuracy in reliability prediction because it considers that the relationship between components, assemblies, and subsystems is deterministic, which is not applicable in stochastic processes such as DFWCS.

Suggestions:

The model should accommodate the limitations of non-reparability of a system to make it suitable for practical applications of SCCS like DFWCS. The approach should be demonstrated for the systems that contain heterogeneous components, which is more likely in case of industrial and safety-related applications.

Sherif Yacoub et al. [109] proposed a model for component based software system to estimate the reliability using Markov model with high level notation based on UML diagram. This approach is supported by component interactions scenarios. This approach build a probabilistic model named Component-Dependency Graph (CDG) by use of scenarios. After that reliability analysis algorithm is constructed from CDG considering system's reliability as a function of reliabilities of its architectural elements. Nevertheless, this approach is worked in three fold for the reliability analysis of a software application: 1) parameter estimation –all the parameter are estimated, viz. scenario-related parameters, component-related parameters, and transition related parameters with the help of various data sources, 2) CDG construction- all the CDG attributes are calculated with the help of control

flow graph principles for the dependencies between components and execution path then after construction of CDG takes place, and 3) reliability analysis algorithm implementation- reliability of the application can be performed as the function of transition and reliability of the components by using this algorithm.. The proposed algorithm is also able to perform sensitivity analysis.

Merits:

This method utilized to test the impact on the overall reliability evaluation of the software system due to uncertainties and variation in the reliability of the specific components, assemblies, subsystems and link between them. This is especially helpful for those systems that are developed from off-the-shelf components partially or fully. The proposed method uses UML due to this for the reliability prediction of the existing design specification can be enhanced quickly to save resources in the later phases of software development life cycle (SDLC). Also, this strategy is utilized to explore critical components, assemblies, and to find out the sensitivity of the specific application reliability dependent to these components.

Limitations:

This approach does not consider cascading failure effect on reliability prediction on the component as well as on the whole software system due to failure dependency of components of a safety critical system. Next, since this approach uses Markov model for the analysis purpose therefore it is not a good practice for analysis of a large scale system due to state space explosion problem. Further, in case of safety critical

system as DFWCS, critical impact of a few scenarios out of many may be more, but they are hardly executed in the proposed approach. Also, this approach does not support time dependency function for application reliability estimation which is very critical in the case of DFWCS.

Suggestions:

We suggest that to overcome the limitation of cascading failure effect, each component's dependency should require a detailed analysis and which should be embedded in this model to obtain an accurate prediction result [65]. Further, SPN method [85] may be used to overcome the limitations of Markov model. It also increases the modeling power to model several features like concurrency, multithreading, etc. which are essentially required for the modeling of DFWCS.

Xujie Jia et al. [110] proposed a reliability model for the reliability analysis of a multistate Markov repairable two-unit series system with zero repair time consideration. This model takes care of multistate of a component and assume that the state of component would be one of the following: 1) complete failure, 2) perfect operation, and 3) minor failure. The different states of the component represented as $(0,\ 1,\ 2)$ or $(F,\ f,\ W)$, where, 0 or F: stands for complete failure; 1or f: stands for minor failure; 2 or W: stands for perfectly working. If the repair time of a complete failure is significantly less than critical value (based on availability of service that can be offered even in case of failure) then system operation does not affected by such failure. Then, based on behavior of the whole system and this assumption,

authors classify the eight states of the two-unit multistate Markov system which are as follows: 1) State classification I- perfectly working system mode (WW), 2) State classification II- system works on minor failure mode(Wf, fW, ff), and 3) State classification III- completely failed system mode (WF, FW, fF, Ff) and modeled by a Markov chain process($X(t)$, $t \geq 0$), which is mathematically defined as defined as

$$X\left(t\right) = j,\ \forall\ j \in \left\{1, 2, 3 \ldots, 8\right\}, for\ the\ system\ is\ in\ statse\ j \tag{3.24}$$

with state space $S = (1, 2, 3, \ldots,\ 8)$. The system model considers the failure rate. After that by use of Markov Model, authors developed another model of the system in which system stops working and failed component immediately gets into the repair process. This new improved model works in, if repair time is less than the critical value, then system can return back in very short time period and so the repair time in this mode neglected as system seen as operational during this period and modeled by a Markov chain process($\widehat{X(t)}$, $t \geq 0$), which is mathematically defined as defined as

$$\widehat{X(t)},\ = j,\ \forall\ j \in \left\{1, 2, 3 \ldots, 8\right\}, for\ the\ system\ is\ in\ statse\ j \tag{3.25}$$

Then after, reliability indexes of a system for both model is calculated and compared.

Merits:

This model consider multi state behavior for a system and its components whereas, many other models have been taken binary state behavior, either working, or failed primarily which gives edge to proposed model for more realistic prediction results. Further, some important reliability indexes, viz. steady-state availability, the mean time to first failure, and instantaneous availability, calculated for the proposed approach which supports to understand the efficiency of the approach.

Limitations:

This approach does not consider complex structure of a system rater it uses very simple structure of two components series system that's why this approach does not qualify to use on safety critical system such as DFWCS which is very complex in nature in terms of hardware and software structure. Further, this approach suffers with state explosion problem due to inclusion of Markov modeling technique in proposed model for the analysis therefore it is not a good practice for a large scale system.

Suggestions:

The approach should be demonstrated for the systems that contain heterogeneous components, which is more likely in case of industrial and safety-related applications. Further, SPN method [85] may be used to overcome the limitations of Markov model. Next, A real case study validation is highly appreciated so that to gain confidence to implement such model for safety-critical systems.

F. Brosch et al. [111] proposed a modeling technique for reliability prediction of a component based software architecture. This technique even include architectural aspect of the software system, viz. the component execution environment and usages profile, explicitly to model and gets more accuracy towards prediction result of the system. The models are turned in to formal analytical model due to proposed technique offers a UML- like modeling notion. Authors have used Palladio Component model to build this work by use of reliability assessment and information propagation. The proposed technique is validated on two case studies with sensitivity analyses and simulation. The case studies investigations show powerful support of usage profile analysis and architectural configuration ranking, together with the work of reliability-improving enhancing engineering strategies.

Merits:

Since Palladio Component model is the base of proposed technique therefore, it has all the merits of its feature and utilize it in software reliability assessment process. The architectural aspects (e.g., the usage profile and execution environment) are also impact on the reliability estimation of a software system, so, addressing such aspects in the proposed technique leads to prediction result more accurate. Further, by the use of tool support, the implementation of the technique gets easier for model transformation into Markov Model and hence for the estimation of reliability. Also, this approach can be used throughout the development process of a software development life cycle (SDLC) and enable software developer to estimate reliability

and go for better architecture alternatives to build the system for critical points of failure.

Limitations:

This model integrates the usage profile and execution environment. The faults/errors from the usage profile will serve a substantial input to the reliability prediction, while in case of safety critical system the probability of occurrence of faults/errors is very rare due to two reasons: (i) SCSs are designed and developed using well set of standards and processes (ii) to capture sufficient amount of faults/errors, usage profile of a very large duration is required.

Therefore accuracy of reliability prediction for such systems is questionable.

Suggestions:

The approach, which is capable to embed the design and development process has to be devised to overcome the limitation. Bayesian approach works on inference principle and hence would be beneficial to embed the design and development mechanism to perform the quantitative assessment of the reliability.

TABLE 3.2: Comparison of Reliability analysis approaches

Existing Techniques	**Modeling Techniques**	**Software failure oriented**	**Hardware failure oriented**	**System (software & hardware) oriented**	**Tool support**	**Illustrating example**	**Simulation/Measurements**	**Real-world application**	**Sensitivity analysis**
Clifford [62], 1967	Mathematical modeling	×	✓	×	×	✓	✓	×	×
M. Faraji et al. [65], 2012	WSPN	×	✓	×	×	✓	✓	×	×
Andre Kleyner et al. [69], 2010	SPN	×	✓	×	✓	✓	✓	✓	✓
G. Ramos et al. [72], 2010	GSPN	×	✓	×	✓	✓	✓	✓	×
Bing Wang et al. [78], 2014	ESPN	×	✓	×	×	✓	✓	✓	×
R. Kumar et al. [82], 2009	MM	×	✓	×	×	✓	✓	✓	×
Zengkai Liu et al. [86], 2015	DSPN	✓	✓	✓	✓	✓	✓	✓	✓
A. Mihalche et al.[93], 2006	SPN	✓	✓	✓	✓	✓		✓	×
K. Krishna et al. [96], 2008	SPN	✓	×	×	✓	✓	✓	✓	×
Chin –Yu Huan et al.[97] , 2005	Goel-Okumoto	✓	×	×	×	×	✓	×	×
Wende Kong et al. [103], 2007	CEGA	✓	×	×	×	✓	✓	✓	×
K. Saravan [107], 2008	Enhanced modeling	✓	×	×	×	×	×	×	×
Yu Liu and Chu-Jie Chenet et al. [108], 2017	Mathematical modeling	×	✓	×	×	✓	✓	✓	×
Sherif Yacoub et al. [109], 2004	CDG with UML	✓	×	×	✓	✓	✓	✓	✓
Xujie Jia et al. [110], 2016	MM	×	✓	×	×	✓	✓	×	×
F. Brosch et al. [111], 2012	MM with UML-notions	✓	×	×	✓	✓	✓	✓	✓

TABLE 3.3: Concluding summary of reliability prediction methods

Category	Reference	Method	Advantages	Disadvantages	Suggestions
Category I:Based on mathematicalmodeling	lifford [62]	Mathematical modeling using reliability physics	**Applied** to heterogeneous components. **Accurate** failure rates for any specific application may be predicted. **Fast** prediction will be possible with actual stress.	**Field** of failure data of components are required. **Requires** strong mathematical background.	**Field** of failure data must be verified. **Extrapolation** techniques may be used. **Operational** profile data of similar systems may be used. **Monte** Carlo Simulation may be used to handle the mathematical complexity.
	Yu Liu and Chu-Jie Chenet [108]	Mathematical modeling with Bayesian method	Uses agglomeration of inspection data collected at multilevel for the dynamic reliability assessment. Due to use of a two stage Bayesian method in recursive manner results more accurate estimation.	Non repairability assumption causes lower estimation accuracy. Since, deterministic assumption is taken about configuration hence, not applicable in stochastic processes.	Accommodate the limitations of non-reparability of a system. Demonstrated for the systems that contain heterogeneous components.
Category II: Based on Markov modeling	R. Kumar et al. [82]	Markov model with constant hazard rate	Non-constant failure rate of the components are considered. Compared to Monte Carlo simulation, mathematical models developed in this paper requires relatively insignificant computer time while achieving high prediction accuracies.	Author used Markov chain which suffers from state space explosion problem.	SPN method may be used to overcome the limitations of Markov model.
	Xujie Jia et al. [110]	Markov model	Addresses multi state behavior for a system and its components. Some important reliability indexes calculated which supports to understand the efficiency of the approach.	Does not consider complex structure of a system. Suffers with state explosion problem.	Approach should be demonstrated for the systems that contain heterogeneous components. SPN method may be used to overcome the limitations of state explosion.
Category III: Based on SPN modeling	Andre Kleyner et al. [69]	SPN modeling	Combines various real life factors (e.g., a user's response time to the warning light, duration of repair, estimated down time, system age). Illustrates the sensitivity of probability of failure on demand. Uses probability of failure on demand or availability to obtain more accurate prediction, instead of utilizing the traditional reliability function. SPN provides a graphical traceability of the solution. More realistic, flexible, and accurate estimate of the system's failure rates.	Authors claim that field data is not required without which attrition function cannot be developed, which is required for analysis of warehouse shipping history for product. Used constant failure rate of the components.	To add the flexibility, SPN method can be effectively combined with traditional reliability analysis techniques, such as Markov chains, standards-based reliability prediction, block diagrams, Weibull analysis, Monte Carlo simulation, etc.

	A. Mihalche et al. [93]	SPN modeling	Results of a co-operational work, gathering mechanical, electronic, and software engineers. Allows reliability evaluation both for n mechatronic systems and for their different sub-systems. Used the estimated parameters instead of the theoretical parameters. Allows reliability is evaluating both for n mechatronic systems and for their different sub-systems. Used the estimated parameters instead of the theoretical parameters.	Jelinski – Moranda model causes less accuracy in reliability prediction. Failed component is immediately repaired, which is not a real scenario.	Detail analysis is required for each component to deal with heterogeneity for more accurate prediction.
	K. Krishna et al. [96]	Stochastic Petri net (SPN) modeling	The reliability estimation of the application development based on the output results causes' significant increase in reliability. Process oriented development.	Prototype building requires effort and resources. Constant failure rate & constant repair rate which limits the reliability prediction accuracy. Only one module taken as a case study, hence interface reliability analysis is missing. No validation.	Non-constant failure rate & repair rate should be consider. Should be demonstrated on a software having multiple heterogeneous modules. Must be validated on real case study.
Category IV: Based on some variants of PN modeling	M. Faraji et al. [65]	Weighed Stochastic Petri Net (WSPN) modeling	Dynamically represents the relationship between infrastructure elements. Addresses infrastructure protection, mitigation, response, and recovery issues. Addresses cascading failure effect. Identifying and quantifying performance of lifeline systems.	Getting Life data for this method is difficult for safety systems. Common mode failure requires much effort.	Instead of using life data, probabilistic data with artificial neural network (ANN) may be used.
	G. Ramos et al. [72]	General stochastic Petri net (GSPN) modeling	Current reliability techniques model disturbances in a probabilistic way; however, they do not model the stochastic response of the power system. System is analyzed under steady-state conditions after the occurrence of disturbances. So, it is not possible to define indicators that include the temporal response of the EIS when sudden disturbances occur.	There should be detection technique which takes a decision that system will be shut down for repair or not based on severity. Cascading fault effect should be considered.	Partial fault detection technique [49] is used. Each and every fault causes EIS shutdown. WGSPN modeling may be used instead of GSPN to identify the component that causes ripple of cascading effect.

	Zengkai Liu et al. [86]	Deterministic stochastic Petri net (DSPN) modeling	Considers human error and implemented on a repairable system to overcome the limitations of earlier approaches. Addresses time limitation for the collected data about subsea BOP failure and malfunction using FTA technique. FMEA technique failed to differentiate the situation of common or severe failure caused by compound failures, whereas DSPN does.	Used constant failure rate of components. Evaluates component reliability only. No methodology to evaluate the interface reliability for evaluation of overall system reliability. Effect of execution environment is missing.	Instead of using constant failure rate we should use Weibull distribution. For accuracy we should consider the reliability of interfaces.
Category V: Based on hybrid modeling	Bing Wang et al. [78]	Fault tree and Extended stochastic Petri net(ESPN)	Using FT based ESPN model for reliability analysis overcomes the limitation of FT. No. of elements required to make FT is more than to make ESPN model. Each transition is associated with its corresponding life distribution function, it can achieve the real time description of reliability analysis. Achieve the dynamic delivery & propagation of reliability/fault information due to the introduction of transition & directed arc.	Not validated on real/test data. Reliability issues of integrated mechanical system and software system is not taking into account.	Proposed methodology can be modified to account the software based systems. Proposed system should be validated to identify the issues.
	Sherif Yacoub et al. [109]	Markov model with UML	Addresses uncertainties and variation and its impact on overall reliability of system. Uses UML due to this for the reliability prediction of the existing design specification can be enhance quickly.	Does not addressed cascading failure effect. Suffers with state space explosion problem. Does not support time dependency function for application.	Each components dependency should require a detailed analysis and embedded it. SPN method may be used to overcome the state space problem.
	F. Brosch et al. [111]	Markov model with UML-notions	Using Palladio Component model in the approach incorporates its features. Address of architectural aspects leads to prediction result more accurate. Can be used throughout the development process and enable software developer to estimate reliability at any phase.	Usage profile of a very large duration is required Occurrence of faults/error is very rare, which is s significant input for reliability prediction	The design and development process of safety critical system needs to be considered in the approach. Bayesian approach is capable to embed this feature.
	Chin –Yu Huan et al. [97]	Goel-Okumoto model	To describe the transitions from the testing phase to the operational phase. Provides useful information to understand the software failure behavior during operation and gives a quantitative analysis of failure distribution in the field operation.	Do not explicitly model the influence of the system usage profile on the control and data flow. Do not consider the reliability impact of a system's execution environment.	Find parameter dependencies for user profile using Stochastic Regular Expressions (SRE).

Category VI: Based on other modeling tech.	Wende Kong et al. [103]	cause-effect Graphing Analysis (CEGA) modeling	Predicting software reliability at the requirement analysis stage could greatly impact on cost, time & hard works. Systematically identify defects in a Software Requirements Specification document. Identifying requirements that are incomplete and ambiguous. A more systematic and clearer path as compared to ad hoc and checklist reading techniques.	Constructing an ACEG for a bulky SRS is very time-consuming. Very difficult to manually determine failure-relevant inputs for a mismatched effect-pair when relevant causes are more than 15.	Use first order predicate logic simplification technique to reduce time at some extent for construction of bulky SRS manually.
	K. Saravan [107]	Enhanced modeling	Knowing the values of software engineering metrics, we can take corrective actions to achieve the required target reliability estimate. Reduce the phase containment of errors.	Initial phases of software life cycle needs to be considered carefully. Since input variables are fuzzy in nature, both fuzzy profiles and the fuzzy rules are not unique.	Use cause-effect graph analysis before the very first of actual development to ensure the correctness at the first phase.

3.5 Safety Prediction: State-of-the-art and Perspectives

There are numerous of conventional methods for safety analysis like FMECA, FTA, HAZOP, SDA, etc., which assists to identification of weak links and defects of system. Table 3.4 [114], [115], [116], [117] shows a comparison of these extensively used safety analysis techniques. In this section, we give an overview of existing safety analysis techniques along with their merits and limitations. The methods to overcome their limitations, with respect to their applicability on NPP systems, are provided.

Karol Rástočný and Juraj Ilavský [112] proposed a method to quantify the safety level of a safety-critical control system. This method uses CTMC analysis to find out hazardous failure rate of a safety control system. Markov chains with

TABLE 3.4: Comparison of safety analysis approaches

Name of Method	**Qualitative**	**Quantitative**	**Top-Down**	**Bottom-Up**	**Complex**	**Failure-oriented**	**Success-oriented**	**Graphical**	**Models sequential actions**	**Models partialFailures**	**Models time dependency**
Expert-opinion	✓	SE	NA	NA	×	✓	×	×	NA	NA	NA
Hazard Indices	×	✓	NA	NA	×	NA	×	×	NA	NA	SE
PHA	✓	×	NA	✓	×	✓	×	SE	×	×	×
FTA	✓	✓	✓	×	✓	✓	×	✓	SE	×	×
ETA	✓	✓	×	✓	SE	✓	×	✓	✓	×	×
FMEA	✓	SE	×	✓	SE	✓	×	SE	×	×	×
FMECA	✓	SE	×	✓	SE	✓	×	SE	✓	✓	×
Cause-consequence	✓	✓	✓	✓	✓	✓	×		✓	×	×
HAZOP	✓	SE	×	✓	✓	×	✓	✓	✓	✓	×
Go Method	×	×	✓	✓	✓	×	✓	✓	✓	✓	✓
Markov Model	×	×	✓	✓	✓	×	✓	✓	✓	✓	✓
SE: Some Extent; NA: Not Applicable;											

more than one absorbing state are contemplated. A case study used to implement the proposed method on the SRCS with a 2-out-of-2 structure which is composed of two independent channels A and B which are identical in hardware architecture and they both control the controlled object (CO). In this technique, two types of CTMC models are proposed, depending on the application. In one CTMC model, it contains only one absorbing state, means that system is not equipped with fault detection and negation mechanism to reach the safe state as well as the system has

two observing states (either hazardous state or safe state).

Hazardous failure rate of system with only one absorbing state (hazardous state,H) can be calculated as,

$$\lambda_H(t) = \frac{\frac{dP_H(t)}{dt}}{1 - P_H(t)} \tag{3.26}$$

where,

λ_H : hazardous failure rate;

$P_H(t)$: probability of the system being in the Hstate;

In other type of model, CTMC contains more than 1 absorbing state. Particularly, in case of SRCS two distinct absorbing states are defined: 1) hazardous state (H), and 2) safe state (S). In this case, the hazardous failure rate of the system may be evaluated as,

$$\lambda_H(t) = \frac{\frac{dP_{HS}(t)}{dt}}{1 - P_{HS}(t)} \cdot \frac{P_H(t)}{P_{HS}(t)} \tag{3.27}$$

And

$$P_{HS}(t) = P_H(t) + P_S(t) \tag{3.28}$$

where,

λ_H : hazardous failure rate;

$P_H(t)$: probability of the system being in the Hstate;

$P_{HS}(t)$: probability of the system being either in the Hstate or in the S state;

From this quantitative value of hazardous failure rate, the SIL level may be checked from the Table 3.5 [113]. According to IEC 61508, Table 3.5 illustrates different SIL Levels. In this table, SIL-4 is highest safety level and SIL-1 is the lowest level. This table shows two modes of operation: 1) Low demand mode- on demand, average probability of failures to run its designated function; 2) High demand or continuous mode- probability of dangerous failure per hour. The safety integrity levels do not associate with failure but with the dangerous failure.

Merits:

This approach can be used to evaluate hazardous failure rate of the system as well as Safety Integrity Level (SIL) of the system. Traditional methods such as RBD and FTA are suffering from the assumption that system can be either in operational state or in failed state completely. But, using CTMC model, the proposed model has the capability to model more generalised system safety properties (failure detection, diagnostic coverage, reconfiguration of a system after detection of a failure, system recovery, etc.) in order to get more appropriate results of the safety analysis [119].

Limitations:

TABLE 3.5: Safety integrity Level (IEC 61508) [113]

Safety Integrity Level	Low demand operation	High demand operation
1	$[10^{-5},\ 10^{-4})$	$[10^{-9},\ 10^{-8})$
2	$[10^{-4},\ 10^{-3})$	$[10^{-8},\ 10^{-7})$
3	$[10^{-3},\ 10^{-2})$	$[10^{-7},\ 10^{-6})$
4	$[10^{-2},\ 10^{-1})$	$[10^{-6},\ 10^{-5})$

The one of the limitations of this method is that it only considers hardware failures. However, safety-critical systems of NPP contains both software and hardware components and the mechanism of software failure is entirely different from that of hardware. Further, the system taken in case study is very simple, does not matches with the complexity of modern systems, like are NPP systems. Hence, it lacks rigorous validation.

Suggestions:

The authors should consider both hardware failures as well as software failures for the safety assessments of the NPP and other safety-critical systems. Experimental and theoretical validation of the method will give enough confidence to use it on safety critical systems of NPP. State-space models may serve both the above purposes.

Yangyang Yuet al. [120] developed a technique for safety sensitivity analysis of the safety-critical systems. This technique is built upon a sensitivity analysis approach for acyclic Markov reliability models and the Markov Chain Modular approach. In this method, the system is divided recursively on the basis of system hierarchy: system contains modules, a module contains either sub modules or components, and sub module contains components (with no fault tolerant

mechanism). The safety sensitivity analysis can be applied to each component and each module of a target system, and thus the overall sensitivity analysis of the system can be achieved. In this paper, a case study used to implement the proposed method on the sensor system. In this approach, N module series system is decomposed in to three types of state space of Markov chain: 1) *Operational state space,* 2) *Fail Safe state space,* 3) *Fail Unsafe state space.* They uses the concept of sensitivity analysis of modules [121], [122] to the component level as below:

Case 1: *Sensitivity Analysis of modules*

The sensitivity of the fail-unsafe probability of Module i is calculated as,

$$I^{Prob}(i \mid t) = \frac{P_{System\_FU\_i}(t)}{P_{M_i\_FU}(t)} - \frac{P_{System\_FU\_i}(t)}{1 - P_{M_i\_FU}(t)} \tag{3.29}$$

where,

$I^{Prob}(i \mid t)$: sensitivity of the fail-unsafe probability of Module i;

$P_{System\_FU\_i}(t)$: the state probability of State FU_i, $\forall i \in (1, 3...N)$;

$P_{M_i\_FU}(t)$: fail-unsafe probability of Module i;

Case 2: *Sensitivity Analysis of Components*

1) If Module i is not further broken down into sub-modules and Module i consists of component k, then the sensitivity evaluation of the system fail-unsafe probability of component k is given by:

$$I^{Prob}_{Component}(k \mid t) = I^{Prob}(i \mid t) \cdot \left(\frac{P_{M_i\_FU\_k}(t)}{P_{k\_FU}(t)} - \frac{P_{M_i\_FU\_k}(t)}{1 - P_{k\_FU}(t)} \right) \tag{3.30}$$

where,

$I^{Prob}_{Component}(k \mid t)$: sensitivity analysis of the system fail-unsafe probability of Component k;

$P_{M_i\_FU\_k}(t)$: the state probability of State FU_k of module i;

$P_{k\_FU}(t)$: fail-unsafe probability of component k;

2) If Module i is further broken down into L layers of sub-modules and the sub-Module h contains component k in Layer$L$, then the sensitivity evaluation of the system fail-unsafe probability of component k is given by:

$$I^{Prob}_{Component}(k \mid t) = I^{Prob}(i \mid t) \cdot \left(\prod_{r-1}^{L} \frac{dP_{M_i(r-1)\_FU}(t)}{dP_{M_i(r)\_FU}(t)} \right) \cdot \left(\frac{P_{M_i(L)\_FU\_k}(t)}{P_{k\_FU}(t)} - \frac{P_{M_i(L)\_FU\_k}(t)}{1 - P_{k\_FU}(t)} \right) \tag{3.31}$$

where,

$M_i(r)$: denotes the sub-module that contains Component k in the Layer r of Module i;

$P_{M_i(L)\_FU\_k}(t)$: the state probability of State FU_k of the sub-module that contains Component k in the Layer L of Module i of the module;

$P_{M_i(r)\_FU}(t)$: fail-unsafe probability of component k in the sub-module that contains Component k in the Layer r of Module i;

Merits:

By the use of this method, we can evaluate whole system sensitivity by performing sensitive analysis of not only target components but also modules as well.

Limitations:

In general, modules are connected in a series-parallel configuration. However, this method works only for the system which modules are connected in a series configuration and hence is not applicable for the safety critical systems of an NPP. The approach works only on hierarchical structures of the components. Validation of this approach is missing.

Suggestions:

This method needs to be more generalised with respect to the arrangements of the components. The technique must be validated on safety-critical systems. The mathematical approach is much appreciated.

Y. Yangyang and Barry W. Johnson [123] introduce two safety-related metrics to evaluate a safety-critical computer-based system. Markov models are used to derive these metrics. In this paper, authors proposed two architectures and evaluated MTTF, Reliability and Safety of these proposed architectures. Addition to safety

metrics, this paper evaluate two other safety-related metrics: 1) System coverability (the ability that a system deals with a failure safety), and 2) Mean Time to Unsafe Failure for constant failure rate (expected time to first unsafe failure). Authors compare and make a decision that which architecture is safer based on the outcome. In first type of the architecture of RTWV system, it contains triplicated modules. In this case, if self-diagnostic routine identifies any kind of fault in one module, the system will act as a duplex system. The second type of architecture contains the redundant module, where if self-diagnostic routines identifies any kind of fault in one module, the system will act as a simplex system.

Quantification of system safety is performed by using Markov chain:

$$S\left(t\right) = \sum P_i\left(t\right) + P_{FS} \tag{3.32}$$

where,

$S\ (t)$: safety of the system or the sum of the probability that a system stays in the operational state;

$P_i\left(t\right)$: probability of the i^{th}operational state of system;

$P_{FS}(t)$: probability of the state which represents fail-safe state of the system ;

Quantification of safety for *architecture system type I* is given by:

$$_{V}C_S - C_V C_S^2 + C_Z^3) + 3(C_S^2 - C_S^3)e^{-\int_0^t \lambda(Z)dz}$$

$$+ 3(C_S + C_V - 2C_S C_V - C_S^2 + C_V C_S^2 + C_S^3)e^{-\int_0^t 2\lambda(Z)dz}$$

$$+ (1 - 3C_S - 3C_V + 5C_S C_V - 2C_V C_S^2 + 3C_S^2 - C_S^3)e^{-\int_0^t 3\lambda(Z)dz}$$

where,

C_V: probability that a voting process recovers given a fault exists in the process;

C_S: probability that a single module recovers given a fault exist in the model;

$P_3(t)$: probability of the state which represents initial state of architecture;

$P_2(t)$: probability of the state which represents system is working in a duplex mode;

$P_{\underline{2}}(t)$: probability of the state which represents system is working in triplex mode;

$P_1(t)$: probability of the state which represents system is working in simplex mode;

$\lambda(t)$: transition failure rate;

Quantification of safety for *architecture system type II* is given by:

$$\begin{aligned} S(t) = \; & P_3(t) + P_1(t) + P_{FS} \\ & = (C_V - C_V C_S + C_S^2) + \left(\frac{3}{2}C_S - \frac{3}{2}C_S^2\right) e^{-\int_0^t \lambda(Z)dz} \\ & - \left(\frac{3}{2}C_S + C_V - C_V C_S - \frac{1}{2}C_S^2 - 1\right) e^{-\int_0^t 3\lambda(Z)dz} \end{aligned} \tag{3.33}$$

Addition to safety metrics, this paper, evaluates two other safety-related metrics using the three-state Markov model for analysing SCS. The three states of Markov model are as following:

1. Operational State: when it is running properly and the probability of the system remaining in the functional state at t is represented as $P_O\,(t)$.

2. Fail-Safe state: when it has actually stopped performing functions, however, the fault has been identified, and the probability that the system is in the fail-safe state at t is represented as $P_{FS}(t)$.

3. Fail-Unsafe State: when it has actually fallen short and the failings have not been managed in a way that ensures the risk-free operation of the system, and the probability of the system remaining in the fail-unsafe State at t is represented as$P_{FU}(t)$.

These three states (out of them two are absorbing states) are mutually exclusive states, and

$$\begin{aligned} P_o(0) &= 1 \\ P_o\,(t) + P_{FS}\,(t) + P_{FU}(t) &= 1 \end{aligned} \tag{3.34}$$

And the additional safety metrics are given by:

1) *System coverability:*

$$SyC\,(t) = P\,(Fail\ Safe \mid A\ Failure\ Exist) = \frac{P_{FS}(t)}{P_{FS}\,(t) + P_{FU}(t)} = \frac{S\,(t) - R(t)}{1 - R(t)} \tag{3.35}$$

where,

$P_{FU}(t)$: probability of the state which represents Fail Unsafe state of the system ;

$R(t)$: reliability of the system;

2) Mean Time to Unsafe Failure for constant failure rate,

$$MTTUF = \frac{MTTF}{1 - SyC_{SS}} \tag{3.36}$$

where,

$MTTF$: Mean Time to Failure;

SyC_{SS}: System coverability at steady state;

Merits:

The proposed safety-related metrics are capable of selecting the best architecture. Using this technique, safety can be quantified at any point of time. This point interpretation has two advantages [124]: 1) the point measure gives the live info of a system's capability to recover; 2) the three-state Markov model has absorbing

states, for that reason, it is simple to change the time dependent state probability $SyC(t)$ by the steady-state probability SyC_{SS}.

Limitations:

The authors assume that the failure rates of the voter and switch are zero and treat them as perfect components during the lifetime of the system. However, this assumption is not practically applicable on SCS of an NPP.

Suggestions:

There must be a proper mechanism to incorporate the failure probability of the voter and switch. Also, failure probability of the interfaces needs to be integrated as they may fail in an undefined manner.

J. Börcsök et al. [125] proposed a paper: "How Safe is my System?" In this paper, the authors quantified many parameters, which are associated with safety. This paper details the criterion Probability of Failure on Demand (PFD). This paper infers the important comparisons and ascertains the PFD-values for various system architectures, which are useful for comparing the safety of different system architectures. The two important safety parameters, discussed are given below:

1) ***Safe failure fraction (SFF) defined as:***

$$SFF = \frac{\sum \lambda_S + \sum \lambda_{DD}}{\sum \lambda_S + \sum \lambda_{DD} + \sum \lambda_{DU}} \tag{3.37}$$

where,

λ_S: safety failure rate;

λ_{DD} : dangerous detected failure rate;

λ_{DU} : dangerous undetected failure rate;

2) *Average probability of failure on demand (1oo1 system),*

$$PFD_{avg} = \frac{1}{T}.\int_0^T PFD\,(t)\,dt = 1 + (\ \frac{\left(e^{-\lambda_D.t}\right) - 1}{\lambda_D.t}) \tag{3.38}$$

T : total operational time;

λ_D : dangerous failures rate;

The first safety parameter will tell the degree of safeness of the overall system, confronting one or more failures and hence helps to take operating decisions in case of an NPP.

The second safety parameter is like defect density model, where safety-related predictions can be made in advance.

Merits:

This paper infers the important comparisons and ascertains the PFD-values for various system architectures, which are useful for comparing the safety of different system architectures. Failure rates of each subsystem are important in the case of reliability not for safety. The authors consider only failure rates of dangerous failures

for the calculations of the probability of failure in low demand mode (PFD), which gives an optimum result of this analysis.

Limitations:

In this paper, the proposed safety parameters have not been experimentally validated. Further, the paper does not discussed its application on SCS. Hence, benefit of its application NPP systems is questionable.

Suggestions:

The proposed method must be demonstrated on a real case study of any SCS, at least if not possible on NPP systems with important findings. The assumptions must be clearly brought out to figure out their validity on the application.

F. Ahmad et al. [126] proposed a method for specification and verification of safety properties along a crossing region in a railway network control. In this paper, author addresses the problem "Resource sharing base on mutual exclusion constraints" in the area of the rail track network. They address above problem in two stages as follows:

1. Specification of safety properties for the model of complex railway crossing having operational safety associated with track & train (occupied, free & block).

2. Develop the control model of the crossing system.

They used Arc-constant coloured Petri net (ac-CPN) to construct the train flow subnet & place- transition Net for modelling of monitors. At last, validation of safety properties in the developed controlled happened using the coverability tree method. In this paper he used two sets, one for trains {TR =(tr_1, tr_2, tr_3, ..., tr_n)} & other for critical region {CR =(tk_1, tk_2, tk_3) }. He defines three safety properties as follows:

$$1) \forall \text{ tr : TR}$$

$$. (\text{tk3, tr}) \mapsto \text{ occupied} \in \text{O} \wedge \text{tk1} \mapsto \text{free} \in \text{F} \wedge tk2 \mapsto free \in F$$

$$\Rightarrow tr \mapsto pass \in P \wedge tk1 \mapsto block \in B \wedge tk2 \mapsto block \in B$$

$$2) \forall \text{ tr : TR}$$

$$. (\text{tk1, tr}) \mapsto \text{ occupied} \in \text{O} \wedge \text{tk3} \mapsto \text{free} \in F$$

$$\Rightarrow tr \mapsto pass \in P \wedge tk3 \mapsto block \in B$$

$$3) \forall \text{ tr : TR}$$

$$. (\text{tk2, tr}) \mapsto \text{ occupied} \in \text{O} \wedge \text{tk3} \mapsto \text{free} \in F$$

$$\Rightarrow tr \mapsto pass \in P \wedge tk3 \mapsto block \in B$$

where,

$$\text{tk1.tk2, tk3} \in \text{CR}$$

tk_i: i^{th} track;

tr_i: i^{th} train;

O: occupied train and track as element of set;

F: free track as element of set;

P: pass train as element of set;

B: block track as the element of a set.

Here, Safety rules 1: shows us if track $tk3$ is occupied then block the track $tk1$ & track tk2. Safety rule 2 & 3: shows if any one or both tk1 & tk2 occupied tk3 must be blocked.

Merits:

The proposed method uses complex rail crossing system along with proper verification of the safety properties in a railway network control. This addresses the limitations of the previous approaches [127], [128], [129]. The technique for the analysis & control of distributed & concurrent system is based on PN, which is a powerful mathematical tool [127], [128].

Limitations:

It gives a qualitative assessment of safety which works in a fruitful manner on non-critical systems, where reliability and safety requirements are not very stringent. However, NPP systems have stringent reliability and safety requirements. This also lacks many more important issues like what happened if any component failed to

perform at the time of need? Authors do not give any idea of sensitivity of component over safety jeopardisation of any components of the system.

Suggestions:

To apply this method on real-time safety critical system, like in a case of NPP, there must be on demand fault detection subsystem which signals to the concerned authority to predict the fault and faulty component identification. The Author should also consider on the reliability of component like train, track, track lever, safety critical software. In this paper author uses PN, which should be used in assessing the performance attribute also.

S. P. Kumar et al. [130] proposed a methodology for building safer software based critical computing systems. He modelled software-safety based on the current software-safety standards, their merits and limitations. In this paper, much emphasis is given on the software element of safety critical systems. This paper compares the different current software safety documents and standards, shown in the Table 3.6. The software safety has been modelled based on the existing software standards along with the information of their advantages and disadvantages. The proposed technology consists of ten tasks: 1) Software safety planning, 2) Safety-Critical Computer System Function Identification and Description, 3) Hazard Analysis, 4) Software Safety Requirements Analysis, 5) Software Safety Architecture Design analysis, 6) Software Safety Detailed Design Analysis, 7) Software Safety Code

Analysis, 8) Software Safety Test Analysis, 9) Software Safety Evaluation, and 10) Software Safety Process Review and Documentation.

Merits:

In this paper, the approach is purely qualitative and hence not much effort is required for mathematical analysis. This approach can be applied to the systems which are not critical to safety and has no mission target.

Limitations:

The technique does not give any quantitative performance indicators and hence is not applicable to SCS of NPP.

Suggestions:

There should be a quantitative approach to estimate the safety attribute or their metrics. There must be a proper validation to ensure its robustness and usefulness in case of NPP SCS. Assumptions must be clearly brought out.

H. Pan et al. [131] proposed a method to model the software safety, and perform computation methods to analyse software safety at the system level, module level and function unit level for the case study of *typical 2 out of 3 system* by using the Markov model.

Merits:

In this paper, a detailed discussion, about different ways of analysis of the system's failure rate is done. It shows software safety analysis at three levels: system level, module level and functional level. The authors also estimate quantitative indicators to measure the safety of the system at different levels.

Limitations:

Due to a use of Markov model, authors use state transition probability as constant. So, this method can be only applied in the case where instantaneous failure rates of system or components are constant. Another disadvantage of this system is that repairable system has a constraint: "system can be restored back to like-new condition after each repair" for this, the present state should be dependent on the previous state but due to Markov model overruled this constraint.

Suggestions:

We suggest that state transition probability should be taken as non-constant to get higher accuracy. In safety, cascading effect is also a major concern, which cannot be addressed in Markov model. For this semi-Markov method may be applied. Further use of PN provides much more modelling power to analyse the insights of the system.

M. B. Swarup et al. [132] proposed an approach to modelling Software Safety. They proposed a framework for risk-free software in view of the McCall's software quality model that particularly recognizes the criteria comparing to software safety in risk-free critical applications. In this paper, authors derived six safety criteria from McCall's software quality criteria as: 1) System Hazard Analysis, 2) Completeness of

Requirements, 3) Identification of Safety-Critical Requirements, 4) Design based on Safety Constraints, 5) Run-time issues management, and 6) Safety-critical testing. Every criteria might be further disintegrated into an arrangement of lower level quality measurements, which are directly quantifiable.

Merits:

It is a standard framework that exhaustively addresses the factors, criteria and metrics (FCM) methodology of the quality models in admiration of risk-free software. These safety quality criteria applied to a prototype road traffic control system (RTCS) and observing the behaviour of its safety violations.

Limitations:

In this modelling technique, only qualitative method is considered. However, in a case of SCS of NPP, the regulatory body is concerned about the quantitative figure to build a confidence level. Other safety parameters like human error, environment conditions are also not considered.

Suggestions:

We suggest that a quantitative method must be proposed with proper validation to provide more confidences towards the safety of SCS of NPP. Human error and environmental condition are also crucial aspects of the safety analysis process. For this proven human reliability analysis methods must be used.

Table 3.6: Comparison of Different Existing Software Safety Documents and Standards [132]

Existing Software Safety Document & Standards	System Safety	Detail Software Safety Process	Software Hazard Risk	Hazard Severity Level
NASA-STD-8719.13A[2]	✓	✓	✓	−−
MIL-STD-882C[4]	✓	×	✓	✓
DO-178B[5]	×	✓	✓	×
JSSC Software System Safety Handbook[6]	×	✓	✓	✓
IEC61508[7]	✓	✓	✓	−−
MISRA[8]	×	✓	×	×
APT Research, Inc.[9]	×	✓	✓	✓

G. Zhou and Huibing Zhao [133] proposed a methodology using FTA and colored Petri net for safety requirements analysis and performance verification of the hot standby system. In this paper, the authors derive safety requirements process. They also described random failure and systematic failure of the hot standby system. For model verification and performance analysis colored Petri net is used.

Merits:

An integrated safety analysis process is described and a CPN model is constructed to help a designer to improve and verify the performance of design scheme which shall satisfy safety requirements. A series of safety requirements including random failure integrity and systematic failure integrity are derived based on hazard identification and risk assessment. It compares some failure and hazard analysis methods. Markov modeling can obtain the safety requirements for hot standby switching failure for accuracy.

Limitations:

As FTA are suffering from the unrealistic assumption that system can be either in operational state or in failed state completely. By using FTA in preliminary hazard identification process, we are unable to do the reconfiguration of a system after the detection of a failure or system recovery of the system. Another limitation is that repair rate of the system has not been taken into consideration, which is applied on SCS of NPP.

Suggestions:

We suggest that instead of using FTA, we may use FMECA method to overcome the stated limitations of FTA in preliminary hazard identification process. However, if we are trying to model a repairable system, PN model may be used.

Abdullah et al. [134] proposed an approach for hazard analysis of the safety-critical system. The methodology comprises of three stages: 1) getting hazards from safety properties, 2) utilizing Fault Tree Analysis (FTA) to break down the conceivable causes of every hazard, and 3) changing every minimal cut-set of FTA into a formal property as far as variables utilized as a part of the formal detail. An Auto-cruise Control (ACC) system for vehicles is utilized as a case study to outline the procedure.

Merits:

The technique is useful to identify the risks of the system. Due to the integration of FTA technique in proposed method, it is advantageous to analyse the possible cause for each hazard.

Limitations:

This paper focuses only on identification of hazards sequentially. However, concurrent hazards are possible in case of SCS of NPP due to its greater complexity. The formal specification is used only at the abstraction level while safety hazards are possible at the detailed level of implementation. Hence, it will not ensure the safety concerns of the overall system.

Suggestions:

The mechanism must incorporate the formal specification at each level of detail. The same must be applied on a real case study to provide enough confidence for its applicability on SCS of NPP.

R. J. Rodrıguez et al. [135] proposed a method that verifies the safety constraints from the early stage of system development life cycle. Due to early verification of safety constraints, this method reduces overall product cost. In this paper, they use UML for system design and Object Constraint Language (OCL) for Specifying safety contract. The verification is done using PN. The approach is validated on a case study to assess the safety of an embedded system, which models a fire prevention system in a hospital building. The proposed model has two steps:1) Safety Contracts Specification: In this work, the authors explore the idea of specifying safety

constraints and guarantees the functional properties of artefact components using OCL invariants within UML models. 2) Safety Contracts Verification: use a subset of the UML behavioural models, namely the UML Sequence Diagram (UML-SD) and the UML State Machine diagram (UML-SM), to express the dynamics of the system. The UML-SD and the UML-SM of can be translated to a combined GSPN state. Now author used the Great SPN tool [136] to validate the model.

Merits:

The proposed method uses qualitative approach for safety analysis. Due to early verification of safety constraints, this method reduces overall product cost. An early verification of safety during a safety-critical system design helps to detect potential safety-related problems. A safety requirement can be specified with a safety contract, which defines the assumptions and guarantees of the functional and safety properties of artefact, assuring its level of confidence. Safety contracts are usually expressed in informal ways, such as descriptive text. In this paper, the authors propose to express them using a predefined syntax to make a transformation to Object Constraint Language (OCL) rules.

Limitations:

Defining the safety contracts in mathematical form would be a very cumbersome process, especially in complex systems like that of NPP. It also requires to have a sound mathematical knowledge and hence is susceptible to use in the industrial applications.

Suggestions:

Integration of informal and different formal models can help in system safety assessment during the design phase, and it deserves further research as it was already stated in [137]. Another suggestion is that to automatize the model transformation stage so that it can be very helpful for extensive evaluation of this approach.

Z. Hong et al. [138] discussed the application of software safety analysis using Event-B, which is a formal method to model the system based on first order and set theory. He shows three different ways to perform safety analysis with Event-B languages and its tool support: (i) Theorem proving (ii) Model checking and (iii) animation. All these three methods are complementary to each other. Theorem proving approach is used to verify the state constraints of the system being modelled. This is an iterative approach with refinement. In each iterated process, three steps will be carried out in sequence: 1) Safety requirements extraction, 2) Transformation of these safety requirements, and 3) Discharging of proof obligation. In model checking, it continuously checks the problem such as developed model satisfies a given specification or not, presence of deadlock, and the other state that causes to the system crash. In another word, we can say that for the safety-critical system it guarantees that the system can transit from dangerous state to safe state. Finally, animation approach is used to view the behavior of the events in a model. It is helpful to visually verified safety requirement as well. In short, procedure of software safety analysis using Event-B in the following steps: 1) Identify the system hazards, 2) Pre-process the requirements, 3) Make the abstract model, 4.a) Analyze with

theorem proving, 4.b) Analyze with model checking, 4.c) Analyze with animation, 5) Refine the model, 6) Repeat step 4 and step 5, till all of the safety requirements are identified and analyzed, and all the defects and weak links of the design are dealt with. This paper also taken a case study and applying software safety analysis based on Event-B to a landing gear extend and retract system.

Merits: The proposed method is capable of overcoming the following limitations: 1) labour intensive, 2) missing human factors impact on safety analysis, and 3) lack of efficiency; of traditional methods (e.g., FMECA, FTA, HAZOP, SDA, etc.). Another advantage of this approach is that it does not have state space explosion problem, usually, formal method suffers with it. Theorem proving, model checking and animation provide systematic approaches to identifying defects and weak links in safety measures and checking whether the safety requirements can be satisfied. Due to refinement process, the method has the ability to change in a model such as extended functionality, updating state, etc.

Limitations:

For safety analysis, it is hard to decide at which level, model checking is to stop, the complexity of the model, and use theorem proving and animation.

Suggestions:

The approach must be modified for quantitative analysis and since the method requires more mathematical background, it can be automized in a form of tool.

Summary of this section with advantages and limitations is given in Table 3.7.

TABLE 3.7: Concluding Summary of Safety Analysis methods

Reference	Method	Advantages	Disadvantages	Suggestions
Karol R'astovcn'y and Juraj Ilavský [112]	CTMC	**Evaluate** hazardous failure rate of system as well as safety integrity level (SIL). **Against** the FTA and RBD, CTMC model more generalised system safety properties	**Only** consider hardware failures **Case** study is very simple. **Does** not validate the proposed technique	**Consider** both hardware failures as well as software failures for the safety assessment. **Complex** system should be taken in the case study. **Proper** validation of the method requires.
Yangyang Yu et al. [120]	Markov Chain Modular Approach	**Perform** sensitivity analysis of target components, modules as well to evaluate whole system sensitivity.	**Works** only for system in which modules are connected in series configuration. **Validation** of this approach is not there.	**May** be more generalised by using sensitivity analysis technique to the system, which contains modules in series-parallel configuration. **Proper** validation is required.
Y. Yangyang and Barry W. Johnson [123]	Markov Model	**With** the help of newly defined safety-related metrics, can compare different architecture. **SyC(t)** gives the live info of a system's capability to recover.	**Assumption** of failure rates of the voter and switch are zero and treat them as perfect components	**Avoid** ideal assumption, may cause unpredicted result.
J. Börcsök et al. [125]	Probability of Failure on Demand (PFD)	**Infers** the important comparisons and ascertains the PFD-values for various system architectures. **Considering** only failure rates of dangerous failures for the calculations of the probability of failure in low demand mode (PFD), which gives an optimum result of this analysis.	**No** proper validation is done.	**Proper** validation of proposed method should be done.
F. Ahmad et al. [126]	Arc-constant colored Petri net (ac-CPN)	**Modeling** of a complex crossing system with verification is given. **Address** and model the complexity associated with multiple switches or crossings. **Due** to PN, this approach is robust and brings a confidence level. **Can** be implemented on concrete railway software.	**Not** able to do a quantitative analysis of safety. **What** happened if any component fails on demand. **Sensitivity** of components on safety is missing.	**There** must be on demand fault detection subsystem which signals the concerned authority if any fault will come in system. **Should** also focus on reliability of components like train, track, track lever, safety critical software
S. P. Kumar et al. [130]	Software-safety standards model	**Purely** qualitative and hence not much effort is required for mathematical analysis. **This** approach can be applied to the systems which are not important to safety and has no mission target.	**Does** not give any quantitative performance indicators.	**There** should be quantitative approach to estimate the safety attribute or their metrics.

H. Pan et al. [131]	Markov Modeling	Detail discussion about different ways of Analysis of the system's failure rate. Giving idea about the software safety analysis includes three levels of safety analysis, system level, module level and functional level. Discuss quantitative indicators to measure the safety of the system at different levels.	The future state of the system is independent of all former states. This method can be applied, when Instantaneous failure rate or constant failure rate assumption is proved to be correct.	State transition probability is constant which should be variable. In Safety, cascading effect is present so we should avoid Markov model.
M. B. Swarup et al. [132]	software safety framework based on the McCall's software quality model	Addresses the factors, criteria and metrics (FCM) methodology of the quality models in admiration of risk-free software. Safety quality criteria are applied to a prototype road traffic control system (RTCS) and observing behaviour of its safety violations.	No quantitative assessment of safety. Other safety parameters like human error, the environment conditions are not considered.	Quantitative assessment should be done. Human error and the environmental condition should be consider
G. Zhou and Huibing Zhao [133]	Colored Petri net (CPN) and Markov model	CPN model is constructed to help designer to improve and verify the performance of design scheme which shall satisfy safety requirements	FTA in preliminary hazard identification process we unable to do reconfigure of a system after detection of a failure or system recovery Repair rate of track side system does not taken into consideration	FMECA method to overcome limitation of FTA in preliminary hazard identification process Should consider repair rate of track side system, so that increase accuracy
A. B. Abdullah et al. [134]	SOFL and FTA	Extracting hazard from safety properties. Integration of FTA advantage to analyse the possible cause for each hazard.	Do not consider any critical or complicated issues, such as concurrent hazards. Only the formal implicit specification is used	Better verification technique still needs
R. J. Rodrıguez et al. [135]	OCL and Petri Net modeling	Saving costs due to addresses safety verification from the early beginning of system development. An early verification detects potential problems that contradict the safety requirements. Embedding all safety related information in a single picture	Defining the safety contracts in mathematical form would be very cumbersome process	Integration of informal and different formal models can help in system safety assessment during design phase
Z. Hong [138]	Event-B modeling	**Traditional** techniques are usually labour intensive, prone to be affected by human factors and lack of efficiency. **Does** not suffer from the problem of state space explosion. **Theorem** proving, model checking and animation provide systematic approaches to identifying defects and weak links in safety measures.	**It** is nondeterministic approach.	**Modified** for quantitative analysis.

3.6 Limitations of existing approaches

The existing state-based analysis techniques have a few impediments that may be clustered into—1) modeling limitations: usually modeling limitations are due to the assumptions we made to ensure model expansibility, which may lead to unsafe estimation, 2) analysis limitations: analysis limitations are due to lack of analysis techniques, 3) parameter estimation limitations: parameter Estimation limitations are because of non-consideration of different system artifacts, 4) validation limitations: validation limitations are because of paying little effort, and 5) optimization limitations: optimization limitations are due to non-consideration of complex interactions between components in the architectural design. These impediments are portrayed in detail as under:

3.6.1 Modeling Limitations

1. In the existing approaches, researchers have used operational profile for the creation of the model. The operational profile is available after testing of the system, and hence, it is not possible to create the model during the early stages of SCSDLC.

2. A practice in which an engineer combines an operational profile and a non-probabilistic specifications to directly produce an analysis-enabled a generative model is tedious, non-intuitive and error prone.

3. The operational profile information that the existing approaches assume available is often just a subset of the available information.

4. Support for discovery and modeling of error states is not clear or accurate.

5. Existing DTMC based models assume that at a time the application can be in one state only which is not valid for today's complex systems.

6. Also, the failure of one component can pass on its impact on other components as well which has not been taken care of anyone of the approaches.

7. None of the approaches has taken into consideration, the nature of the interface between the components, as there is a possibility that components may be distributed with the advanced technologies.

8. The architectural style of different components of the same application system may be different, for which we suspect to fit the common safety approach on all the components. Also dynamically i.e. when application operates its architecture also changes dynamically.

3.6.2 Analysis Limitations

1. The safety of the CBS is a function of the safety of each of its subsystems and their connectors. There should be a mechanism through which the impact of the change of any of the component's or connector's safety on the system safety can be found to ensure the target safety requirements of the system.

2. In some approaches, Hidden Markov model has been used when it is difficult to have the surety of next probabilistic transition. But in this case, the transition matrix and observation the probability matrix (which represents the probability of observing event in a particular state) has been initialized randomly, which may not be accurate.

3.6.3 Parameter Estimation Limitations

1. The safety of the CBS, based on Markov chain, is a function of transition probabilities in between the states of Markov chain. In the existing approaches these have been assumed analytically, and hence the accuracy of the predicted values is not guaranteed.

2. The system-level model can be analyzed using traditional DTMC analysis [20], whose complexity is $O(n^3)$, where n is the number of states. Generally, large complex software systems have thousands of states, which could be very expensive to solve the DTMC model.

3.6.4 Validation Limitations

The less effort has been paid to validate the predicted safety, based on an architectural design with the estimated safety/risk/hazard, just before product release to ensure the correctness of the predicted methodology so that, it can be applied to the future projects.

3.6.5 Optimization Limitations

Safety prediction based on architecture can optimize if we success to optimize the system architecture. Sometimes it is noticed that architects design the system in a complex manner, full of tight coupling and low cohesion, which is a poor quality attributes of architecture.

3.7 Conclusion

Reliability and safety prediction approaches have been intensively studied in the past, and several research papers have been published that address reliability and safety analysis issues. We have attempted to bring out the major contribution made by the discussed research and identified strengths and weaknesses of the models proposed and discussed in this chapter. Many of them lack to address limitations that discussed above and hence to make such a model understandable some additional information may not be available. We find sufficient opportunity and scope for improving and perfecting methods and models to make them more useful for researchers and practitioner. Our extensive state-of-art review brings out the above said interesting observation and some possible ways of dealing with them. We have made every effort to provide an orderly state-of-the-art review on safety critical system's reliability and safety analysis some alternate points of view as per our understandings. A significant number of related studies and results have been summarized to explore the current challenges that need to be tackle.

In this Chapter, we try to explain the fundamental issues involved in predicting the reliability of SCCS as case study DFWCS. .Further, we also try to explain the fundamental issues involved with predicting the safety of SCCS. It is trusted that this survey is likely to be beneficiary to all those contemplating the utilization of SCCS by emphasizing several of the issues and proper strategies that ought to be considered in the development of critical applications. The review identifies the limitations of existing methods with respect to its applicability on the SCCS as well as its applicability on the DFWCS (a SCCS). The suggestions to modify the existing methods are also mentioned, to overcome the limitations. Various tables given in this chapter which provide a comparative study of several existing reliability analysis techniques, whereas some of tables provide their concluding summary based on various methodology used that includes advantages and disadvantages, along with the suggestions. These can help the researchers, academicians and practitioners to address the following questions.

1. What kind of modeling techniques are used in an assessment process under consideration?

2. What are the advantages of using a given reliability assessment technique?

3. What are the limitations of the existing techniques?

4. How to overcome the limitations?

5. How to know the feasibility of applying existing techniques to a specific system.

We have considered throughout our case study how the said models could be applied. We have shown that heterogeneous and synthesized approach in form of suggestions. Then we can work out an efficient reliability prediction approach for a SCS.

This literature review has helped us in formulation of in formulation of research problems identification of planned solution strategies for the same, being explored and elaborated in the next chapter, Chapter 4.