Educational Administration: Theory and Practice
2024, 30(4), 2945-2957
ISSN: 2148-2403
https://kuey.net/

Research Article

A Stochastic Analysis of Computing Models in Computer System Reliability Based on Hardware and Software

Yadav Vikal[1*], Milind[2]

[1*,2]Department of Computer Science and Engineering, SCRIET, Chaudhary Charan Singh University Meerut, vikalyadav512@gmail.com, milindccsu@yahoo.com

Citation: Yadav Vikal.et al. (2024), A Stochastic Analysis of Computing Models in Computer System Reliability Based on Hardware and Software..., *Educational Administration: Theory And Practice*, *30*(4), 2945-2957, Doi:10.53555/kuey.v30i3.1962

ARTICLE INFO

ABSTRACT

Computer systems have become integral parts of modern society, playing critical roles in various sectors such as finance, healthcare, transportation, and communication. As these systems become increasingly complex, ensuring their reliability becomes a paramount concern. Computer predicts have emerged as indispensable tools for assessing and predicting the reliability among such systems. The intention of this study article is to offer an extensive analysis the ones that role moreover effectiveness that of computer models across evaluating computer system reliability. Through an exploration of various modeling techniques, case studies, and challenges, this paper seeks to highlight the significance of computer models in enhancing our knowing of system reliability and promoting efficient decision-making processes. A technique's reliability is a system's capacity to consistently operate as meant under an identified set of requirements.

This article covers dependability modeling approaches and examines their positive and negative features. Among the models examined are basic stochastic models, decomposable stochastic models, and structure models. Eliminating time-dependence, structure models system dependability depending on the topological character within the framework. Basic stochastic models directly leverage the attributes within the core stochastic treatments, nevertheless broken-down Models analyze deeper entities through their sections. Complex system analysis can be made easier by the realistic foundation for reliability analysis delivered by Petri nets and dataflow graphs.

Keywords: dataflow graphs, stochastic models, decomposition-aggregation approach, and reliability.

1. Introduction

Most studies have taken into account a portion either a software subsystem, a hardware subsystem, or the system itself in order to simulate system reliability. Many software and hardware reliability models have been developed during the last few decades in an effort to address failures in software and hardware subsystems, respectively, from distinct angles and taking into account numerous important applications. In most research, Software and hardware components are taken for granted as autonomous since the relationships between them are frequently disregarded in an effort to streamline mathematical formulation. This presumption might not be accurate in practice, though. Computer models, including analytical models, simulation models, and hybrid models, have gained prominence as essential tools in assessing system reliability. This paper delivers into the various aspects of using computer models for analyzing computer system reliability, providing insights into their benefits, challenges, and applications. [7-12]

The complexity of equipment and settings has been on the rise as computer systems strive to operate as efficiently as possible. A single indicator of a system's effectiveness is dependability, which is the likelihood considering the setup successfully completes it planned purpose to serve a predetermined amount the period spent beneath predetermined ecological circumstance. In case a suitable quantitative framework for the system can be created, reliability can be calculated using that model. As computer systems attempt to function as effectively as possible, the complexity of the tools and environments has increased. Dependability, or the likelihood that a system will carry out its intended duty successfully for a controlled

setting for a certain amount of time, is one sign of a system's efficacy. Reliability can be determined using the system's mathematical model, if one can be developed.
As computer systems attempt to function as effectively as possible, the complexity of the tools and environments has increased. Consistency, or the likelihood that a system will carry out its intended duty successfully regarding a set duration term predetermined ecological circumstance, is one sign of a system's efficacy. Reliability can be determined using the system's mathematical model, if one can be developed. A topological reliability model takes system structure at a given point in time into account for determining dependability. The entire system reliability model, which takes into account the software and hardware subsystems, was only partially defined in a few researches. A dependability model that is reliant on time, on that other hand, treats The structure's features state as a stochastic technique, with dependability being determined as a process-level operational metric. Within the parts that follow, Let's talk about a few particular exemplary models that are either structural, stochastic, or both. [6]

2. Structure Models

Stochastic networks frequently serve as tools for the examination of intricate systems; their arcs indicate each of the elements of the system. As a consequence, a likelihood that what a part does is equal to the statistical the fact that the relevant network electric performs. A network's trustworthiness is determined by the likelihood that the line connecting the origin and exit nodes is made up only of functional loops. The dependability issue is changed with this network model to the difficulty of determining the paths through the connection as well as the resulting likelihood [20]
A dependability network's structure is characterized by structural functions. Think of a system with r members, such as A_1, A_2....A_z. connect a state variable y_i in a way that

$$y_i = \begin{cases} 1 & \text{If } A_i \text{ is operating} \\ 2 & \text{if } A_i \text{ Has not succeeded} \end{cases}$$

In a similar way, establish the status parameter y_i.

$$x_i = \begin{cases} 1 & \text{Provided that the system functions} \\ 2 & \text{Should the system malfunction} \end{cases}$$

It is now possible to design an operation ϕ such that

$$x = \phi(y_1 , y_2 , ...y_r); \quad (1)$$

The expression "structure functions" characterises this. Structure functions that are homogenous (sometimes referred to as coherent) may be used to depict reliability networks. The configuration of the function is present, for instance, within a network of sequence with r elements.

$$x = \prod_{i=1}^{r} y_{1,} \quad (2)$$

When at least one component in a parallel r-component network having the structural function is required for the system to succeed,

$$x = 1 - \prod_{i=1}^{r} (1 - y_1) \quad (3)$$

Since a unit's dependability is determined by the likelihood that the matching state statistic y will take on The integer 1 when every aspect of the network runs apart from the others, the reliability of a network is calculated by substituting the y's in the structure function with the corresponding reliabilities. Let q_i for instance, represent the component A's (i = 1, 2.....m) reliability. Then, accordingly, the reliability of the concurrent (paral) and sequential (ser) structures is ascertained.

$$\left.\begin{aligned} R_{scr} &= \prod_{i=1}^{r} q_i \\ R_{paral} &= 1 - \prod_{i=1}^{r} (1-q_i) \end{aligned}\right\} \quad (4)$$

One can use minimal paths or lowest deletions of the foundation graph to find the basic functions of a given trustworthiness networks. for an in-depth analysis of structural function characteristics and how crucial they are in establishing the reliability of intricate structures. Graph theoretic topics like path, state, and cutset enumerations are commonly utilized in stochastic network techniques for dependability computation. The fact that the number of operations increases linearly via network growth is an important bug in present scanning algorithms.

If the network consists of n nodes, then 2" states need to be considered. Using path enumeration techniques, Inclusion-Exclusion Hypothesis for The likelihood is applied to determine the probabilities from the identified pathways. Let Nr be the likelihood that precisely s of the n pathways (A1,A2,..... An) Connecting Along with the outcome of the node, the inputting node k are functional. It is now likely that at least one of the channels functions are supplied by.

$$R1 = N_1 - N_2 + N_3 - .....\pm N_n \quad (5)$$

Furthermore, it's probably the case that at least m of the n routes operate are provided by

$$R_u = \sum_{s=u}^{v} (-1)^{s-u} \begin{pmatrix} s-1 \\ u-1 \end{pmatrix} N_s \quad (6)$$

Many of the elements in (5) are cancelled as a result of the inclusion-exclusion principle, and their presence originally makes the issue larger. An algorithm founded on the idea of hegemony in the network's foundational topology that uses just the no cancelling terms of (5). For further developments of this method and associated outcomes. For a superb analysis of network dependability that makes use of special structure and dominance theory. The decomposition method and the usage of upper and lower bounds are further methods for simplifying the issue. Other algorithms have been discussed as well.

In digital networks, parallel processing is a unique problem that regularly arises and has drawn a lot of attention. The basis of this technique is the concurrent execution of many computing steps. Many kinds of graph theoretic methods have been tried and failed to resolve this challenge. The most effective strategy to date has been reduction, which considers necessary consecutive curves as curves arranged in order. Should just a portion of the comparable curve is needed, the set of parallel curves is replaced by just one curve that has faithfulness equal to the concurrent set.

3 Simple Markovian Models

A complex system is one that has a large number of interdependent components. Numerous important contemporary applications, like computing and communication systems, are made up of numerous hardware and software components. Generally speaking, the overall system could be the result of one or more component failures. Three categories are used in this study's classification of system failures. [22]

Consider a system the fact consists of two halves, one of is operating and the other of which is in reserve. Let $\lambda_2(\lambda_2 \leq \lambda_1)$ represent the (steady) inability speed of the part that is functioning and λ_1 represent how frequently the back-up portion fails. Under this assumption, there are two methods to assess dependability (constant failure rate implies exponential life spans). Starting with the simpler approach, let's identify the events that facilitate the system's smooth functioning between (0, h). Let $Q_n(z)$ represent the likelihood that n(n = 0, 1, 2) is the integer of inability components during (0, r).

Clearly, there is a probability density

$$Q_n(z) = \exp(-\lambda_1 z)\exp(-\lambda_2 z) = \exp[-(\lambda_1 + \lambda_2)z] \quad (7)$$

Both the operational and the standby components may fail when one component does. In the latter scenario, the backup component quickly replaces the failed component. So, we obtain

$$Q_n(z) = \exp(-\lambda_1 z)[1 - \exp(-\lambda_2 z)] + \int_0^z \lambda_1 \exp(-\lambda_1 Z)\exp(-\lambda_2 Z)\exp[-\lambda_1(z-Z)]dz$$
$$= [(\lambda_1 + \lambda_2)/\lambda_2]\{\exp(-\lambda_1 z)\exp[-(\lambda_1 + \lambda_2)z]\} \quad (8)$$

Now, trustworthiness R (z) may be found as

$$R(z) = Q_0(z) + Q_1(z)$$
$$= [(\lambda_1 + \lambda_2)/\lambda_2]\{\exp(-\lambda_1 z) - (\lambda_1/\lambda_2)\exp[-(\lambda_1 + \lambda_2)z] \quad (9)$$

The duration until the first failure, which comes after the first failure and had an exponential shape with an error rate of (λ 1 + λ 2), and the duration until the second failure, which comes after the first malfunction and has an explosive distributions with a failure rate of λ1, make up the system life. This results in a likelihood ratio for the entire system's life span.

$$f(z) = \int_0^z (\lambda_1 + \lambda_2)\exp[-=(\lambda_1 + \lambda_2)Z]\lambda_1 \exp[=[-\lambda_1(z-Z)]dz$$

$$=(\lambda_1 / \lambda_2)(\lambda_1 + \lambda_2)\{\exp=[(\lambda_1 z) - \exp[-(\lambda_1 + \lambda_2)z]\} \quad (10)$$

This justification can also be used to systems Utilizing hybrid N-tuple adaptable redundant work, or mix NMR. S + N independent components, of which S are active and the other N are on standby, make up a hybrid NMn system. For the system to work correctly, it needs at least u components. One of the system components that comprise fault-tolerant computer systems is the hybrid NMR.

3.1 Fault Coverage

To make the model more realistic, the coverage parameter has been incorporated to the creation of computer systems that can withstand faults. For instance, when an online component fails in a hybrid NMn system, it might not be able to swap in a spare part quickly enough to heal. If it occurs, the error is considered to have been discovered. The coverage parameter is the likelihood that a fault can be repaired.

Think about enhancing the one unit standby system that was previously proposed with the coverage functionality. Let λ_1 and λ_2 represent The reserve and on-line sections' ongoing rate of breakdowns, as well at and. Let d¬1 and d2 stand for the system's odds of recovering in the event that the online and standby units fail, respectively. A straight Markov process approach might make sense in this case. [17]

For the probability Q¬n (z) (n = 0, 1, and 2) We acquire the difference-differential solutions since we know that the system comprises n failed units at time z.

$$Q/_0(z) = -(\lambda_1 + \lambda_2) Q_0(z)$$

$$Q/_1(z) = -\lambda_1 Q_1(z) + (\lambda_1 d_1 + \lambda_2 d_2) Q_0(z)$$

$$Q/_2(z) = [(1 - d_1)\lambda_1 + (1 - d_2)\lambda_2] Q_0(z) + \lambda_1 Q_1(z) \quad (11)$$

Q1(0) = 1 for n' = -0 and = 0 in all other instances as the initial condition. Laplace changes can be utilized to solve the array of problems (11) and generate.

$$Q_0(z) = \exp[-(\lambda_1 + \lambda_2) z]$$

$$Q_1(z) = [(\lambda_1 d_1 + \lambda_2 d_2)\lambda_2]\{\exp[-\lambda_1 z] - \exp[-(\lambda_1 + \lambda_2)z]\}$$

$$Q_1(z) = [(1 - d_1)\lambda_1 + (1 - d_2)\lambda_2] \exp[-(\lambda_1 + \lambda_2) z] +$$

$$+ (\lambda_1 / \lambda_1)(\lambda_1 d_1 + \lambda_2 d_2)\{\exp(-\lambda_1 z) - \exp[-(\lambda_1 + \lambda_2)z]\} \quad (12)$$

Note that the probability density of system life is also $Q_2(z)$. Employing the coverage parameter referred to as "equivalent"

$$d = (\lambda_1 d_1 + \lambda_2 d_2) / (\lambda_1 + \lambda_2)$$

And keeping in mind that $Q_0(z) + Q_1(1)$ provides the reliability R (z), we obtain

$$R(z) = \exp[-(\lambda_1 + \lambda_2) z] + [z(\lambda_1 + \lambda_2) / \lambda_2] x$$
$$x \{\exp(-\lambda_1 z) - \exp[-(\lambda_1 + \lambda_2)z]\} \quad (13)$$

The Laplace transforms (LT) of $Q_0(z)$and$Q_1(z)$ can be used to directly Determine the normal period of breakdown. In case that is all that matters. If $\phi_n(\theta)$ represents the LT of $Q_0(z)$ and $q(\theta) = \phi_0(\theta) + \phi_1(\theta)$, the anticipated framework's life E[L] is provided below.

$$E[L] = \int_0^\infty R(z)dz = \lim_{\theta \to 0} q(\theta) \quad (14)$$

$$=[1 / (\lambda_1 + \lambda_2)][1 + (\lambda_1 + \lambda_2) d/\lambda_1] \quad (15)$$

The dependability characteristics of the hybrid NMR system mentioned before with a coverage parameter can be obtained using a similar way. For the expression for E [L], this is obtained using the conditional distribution justifications shown.

In the preceding case, The range of likelihood is known, as we predicted. Truly, in life, estimation is challenging. Below is an indirect estimating approach that provides more insight into the coverage phenomenon.

A Markov model that can be used to find errors looks like this one. The model consists of five states. The states are in this order: A is the functional state; B is the good state; E is the error state; D is the identification

state; and F is the unsuccessful state. Let λ represent the rate of error detection and â represent the rate at which A creates errors. Assuming that the defects are intermittent in nature, the approach alternates between the benign condition B, where no mistakes are created, and the active state A.α (for A→B), and β(for B→A). The error state E can also be used to identify the error. Allow the shift in rates to E→D and E→F, respectively, be qγ and mγ (m = 1 - q). For constant transition rates, we have a Markov model and exponential residency durations. Differential equations control the process.

$$m_A'(z) = -(\alpha + \gamma + \delta)m_A(z) + \beta m_B(z)$$
$$m_B'(z) = -\beta m_B(z) + \alpha m_A(z)$$
$$m_E'(z) = -\gamma m_E(z) + \lambda m_A(z)$$
$$m_D'(z) = -\delta m_A(z) + q\gamma m_E'(z)$$
$$m_F'(z) = m\gamma m_E(z) \qquad (16)$$

With the original circumstance (0) = 1. By simplifying Laplace transforms in the conventional way, we obtain

$$\phi_D(\theta) = (1/\theta)[\delta + q\gamma\lambda/(\phi + \gamma)]\{(\phi + \beta)/[(\phi + \beta)(\theta + \alpha + \lambda + \delta) - \alpha\beta]\} \qquad (17)$$

The likelihood that the defect will eventually be found before the system crashes is clearly given by $\lim_{z\to\infty} m_D(z)$. As a result, the process's coverage probability can be calculated using this probability. We provide an estimate of the coverage probability using Laplace transform properties.

$$d = \lim_{z\to\infty} m_D(z) = \lim_{\theta\to 0} \theta\delta_D(\theta) = (\delta + q\lambda)/(\delta + \theta) \qquad (18)$$

3.2 Systems with Repair

We have only considered unmaintained systems in the aforementioned examples, when a malfunctioning component is simply replaced rather being fixed. These characteristics of a suitable Markov process model are present in pure death processes as they may be examined as Markov processes and with minimal likelihood with qualifiers considerations. But when the item's upkeep option is used, only Markov methods succeed and conventional reasons collapse. Function—apart from a few simple system setups. Here are a few instances of these issues.

Take the above-mentioned single-unit failover device using the upkeep feature into mind.. Let λ_1 and λ_2 represent the on-line and standby components' respective constant failure rates. It is imperative that the repair time of a damaged item, regardless of its online or standby mode usage, has an average value of and spreads linearly. 1/μ. We now have the difference-differential equations regarding the likelihoods $m_n(z)$ (n = 0, 1, 2) that At times, the framework z has n failed modules.

$$m_0'(z) = -(\lambda_1 + \lambda_2)m_0(z) + \mu m_1(z)$$
$$m_1'(z) = -(\lambda_1 + \mu)m_1(z) + (\lambda_1 + \lambda_2)m_0(z)$$
$$m_2'(z) = \lambda_1 m_1(z) \qquad (19)$$

$m_n(0)$ is initially referred to as 1 in the case that n is larger than 0 and identical to 0 in the other condition. If not, discover becomes 0. With the use of LT, these equations can be solved. When a lacking coverage feature arrives to the model with coverage likelihood, linear equations arise of d_1 and d_2.

$$m_0'(z) = -(\lambda_1 + \lambda_2)m_0(z) + \mu m_1(z)$$
$$m_1'(z) = -(\lambda_1 + \mu)m_1(z) + (\lambda_1 d_1 + \lambda_2 d_2)m_0(z)$$
$$m_2'(z) = [(1 - d_1)\lambda_1 + (1 - d_2)\lambda_2]m_0(z) + \lambda_1 m_1(z) \qquad (20)$$

3.3 Availability

Another reliability trait that indicates the likelihood over the long term that the system will be usable is availability. Evaluating the trouble-free single-unit contingency arrangement, the way it works is explained in (16), can be observed, when both units split down, the system collapses. Assume once the damaged components are fixed, the system becomes functional once again. After then, state 2 (containing two failed units) won't be able to absorb more. Knowing that in the event that two of them fail the steady state equations provide the long-term likelihood mn (n = 0, 1, 2) ensuring that n fall items exist. At that point of detection. The two units undergo replacement in tandem, and the rate of μ causes the repair times expand exponentially. [18]

$$(\lambda_1 + \lambda_2)m_0 = \mu m_1$$
$$(\lambda_1 + \mu)m_0 = (\lambda_1 + \lambda_2)m_0 + 2\mu m_2$$
$$2\mu m_2 = \lambda_1 m_1$$

Solving these equations using the extra condition as guidance $\sum_{0}^{2} m_{n=1}$ we get

$$m_0 = 2\mu^2/J$$
$$m_1 = 2\mu(\lambda_1 + \lambda_2)/J$$

$$m_2 = \lambda_1(\lambda_1 + \lambda_2)/J \quad (21)$$

Where

$$J = 2\mu^2 + (2\mu + \lambda_1)(\lambda_1 + \lambda_2)$$

Accessibility of the framework

$$A = m_0 + m_1 = 2\mu\,(\mu + \lambda_1 + \lambda_2)/J \quad (22)$$

The analysis of maintained systems and the analysis of queued systems are quite similar, as was shown above. Failures arise from the client's arrival, and service times represent the amount of time needed for repairs. The literature has a large number of publications that make use of this commonality. In several of these works, non-exponential versions of broken parts or turnaround times are being factored into effect.

3.4 NMR Systems With Voting

"Voters" In the previously mentioned NMR systems, data coming from the simultaneous parts are juxtaposed to ensure that every one of them are operating. Consider on a majority vote triple multilateral redundancy (TMR) system. The rejection rate is always elevated of λ_1 as well as maintenance frequency of μ for the three parallel components.

Let λ_2 be the voter failure rate λ_3 in the case that all three components are operational and â, in the event that Three of them have tails. Now, an array (n, m) indicating the vote's circumstances and the quantity of parts that failed. (0 for a working electorate and 1, or a failed electorate) must be utilized to define the status space.

The likelihoods $Q_{nm}(n=0,1,2;m=0,1)$ fulfill the formulas that

$$Q_{00}'(z) = -(3\lambda_1 + \lambda_2)Q_{00}(z) + \mu_1 Q_{10}(z)$$
$$Q_{10}'(z) = -(2\lambda_1 + \lambda_3 + \mu_1)Q_{10}(z) + 3\lambda_1 Q_{00}(z)$$
$$Q_{20}'(z) = 2\lambda_1 Q_{10}(z)$$
$$Q_{11}'(z) = \lambda_3 Q_{10}(z) \quad (23)$$

$Q_{nm}(0)$ only equals 1 with the starting condition either 0 or if n = 0 and m = 0. As stated earlier, the parallel components' efficacy determines the voter's effectiveness. This dependence forces us to employ a broader spatial state. If both of the regions and The quartet of concurrent elements are present, voter independence allows their study to be divided into two separate systems. Expanding the state space is necessary to give information on every element in a larger network with interconnected subsystems. As such, one must also deal with the issue of an increasing state space when applying a Markov process model on a big network. Using a decomposition-aggregation technique, which looks at subsystems first and then aggregates the results to offer the consortium's broad view, is one way to get around this problem. The following section goes over it. We showcase three durability forecasting techniques for big reliable machines to wrap off this section.

4. Decomposable Stochastic Models

Queuing networks have been successfully analyzed using the decomposition-aggregation solution approach. The strategy's principal premise is to divide the network into more manageable, smaller sub-networks and then combine the outcomes into a bigger network of smaller networks. The results obtained can only be approximated if the sub-networks are not independent or, to use the terminology of a Markov process, generate discontinuous sets of states. This is because interactions between the states in separate sub-networks must be avoided for collecting the data. On the reverse hand, when there are little interdependencies among the networks of subnetworks the method has demonstrated value in queuing networks. A simplified reliability analysis that might be utilized to demonstrate the advantage of the case III method against stability modeling is used as an instance of this technique hereunder computer networks.

To compute the coverage likelihood, take account of the earlier scenario. A five-state model— statuses categorized as active (A), benign (B), error-prone (E), detectable (D), and fail (F)—was taken into consideration in those circumstances. At the moment, we had only thought regarding a single package. Let's imagine there are two components, and at least one of them must be in working order for the system to succeed. While there are two units, there are two types of failures that can happen: one uncovered failure brought on by an undetected defect, and another uncovered failure brought on by the absence of operational squads while the covered faults are present. Thus, the different states of the model consist of: A number between 0 and lA denotes the operation of both units; a number between lB and lD denotes the benign condition of the unit; and a number between lE and lF denotes the failure state resulting from one or more uncovered failures.[1,11,19]

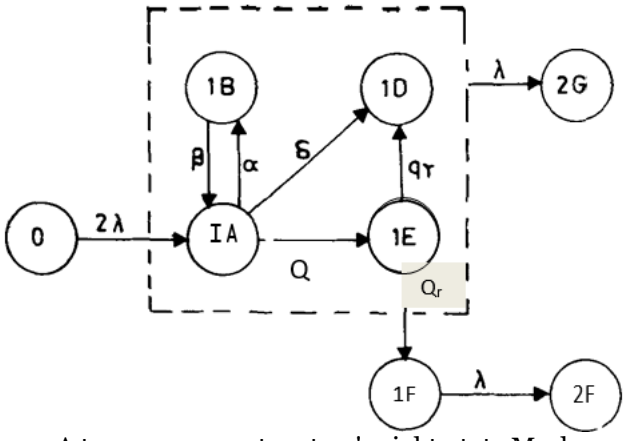


Figure 1 A two-component system's eight-state Markov model

Unit 2F is the break state that results from the lack of working units, while Unit 2G is the fail condition that emerges from both units' undetected failure. A model of Markov chains for clear statements of chance that is akin to the one below whose movements are governed by (18) Qn(z) (n —— 0, lA, 1E, 1D, I F, 2F, and 2G) can be constructed with the notion that the transition rates are fixed. However, the benign state lB is left out for simplicity's sake. By investigating a 5-state methodology as a successor to the first 8-state procedure and unifying states lA, 1B, IE, and lD into a single state, let's call I G, you can approach what was previously indicated system. Figures 1 and 2 illustrate this.

Observe the fact that the module subject to Equation (18) is the same as the fault detection system inside the square box in The first figure If the framework depicted in Figure 2 is to mimic the system constituted in Figure 1, then the rates of transition from I G to 2G and 2F in Figure 2 should match those in Figure 1. It appears that these do not have to be constants in order for the modified system to be a time-homogeneous Markov process. They are easily distinguishable from one another. However, as in the fault detection model, this could be roughly finished by investigating part lG independently and determining the rate at which the subsystem moves from part IG to state IF, in order for the mean occupancy duration of the system in IG to correlate with the sum of its imply time at rest in states lA, lB, 1D, and 1E. We'll ignore state l B and set (α = β = 0) to keep things effortless. It's obvious the average time spent in presence in state n can be determined by

$$\lim_{\theta \to 0} \int_0^{\infty} \exp(-\theta z) m_n(z) dz$$

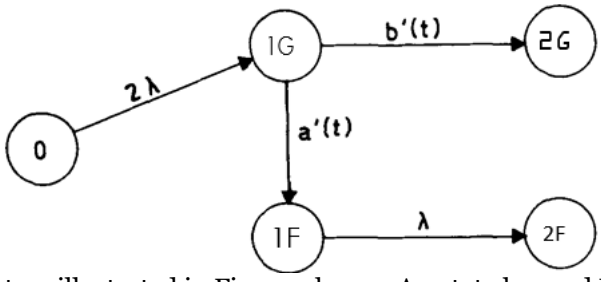


Figure2. The system illustrated in Figure 1 has an A5-state lumped Markov model.

As in 1A, 1D, and 1E, the aggregate mean residency duration is determined as

$$\eta = [\lambda(\gamma + \lambda + p + \delta) + (\delta + qp)]/[\lambda(\gamma + \lambda)(\delta + p + \lambda)] \quad (24)$$

Assume that the pace at which l G and 1F transition is a constant, a. Next, the process's average length of stay in the state lG, represented by r, is calculated as

$$r = 1/(\alpha + \lambda) \quad (25)$$

Comparing equations (24, 25)

$$\alpha = p\gamma\{\lambda p / [\lambda(\gamma + \lambda + p + \delta) + \gamma(\delta + qp)]\} \quad (26)$$

Now, the entire system may be analyzed as a simple time-homogeneous Markov chain process using α standing as the unchanging change phase over 1G and IF.

As an alternative, it is possible to establish a time-dependent transition rate α'(z) between 1G and 1F and calculate α'(z) using the relation

$$[Q_{1A}(z) + Q_{1B}(z) + Q_{1D}(z) + Q_{1E}(z)]\alpha'(z) = p\gamma Q_{1E}(z) \quad (27)$$

This leads inexorably to an inconsistent Markov process, and its study is very complicated and necessitates a combination of explicit answers for the likelihood in (27), in which case a different set of calculations to explicitly learn them needs to be used and the calculated figure is not needed.

The ability to apply suitable solutions and sub-network techniques is one of the benefits of the decomposition-aggregation strategy. The Hyperbolic Computerized Quality Forecaster, or The HARP model is a model including components for both evaluation and exercise. The aforementioned hybrid model partially addresses the main shortcomings that plague ultrahigh fidelity forecast models, such CARE III (as well as ARIES, SURF, and CAST described in the section above). Dream with a computer system that can endure errors., like the one previously mentioned, which is split into sub components for fault-occurrence and fault-handling, as an example of this technique. The observation that a system's fault-occurrence behavior consists of comparatively uncommon events, while its fault-handling behavior consists of extremely frequent occurrences, lends credence to this breakdown. Using differential equation systems, the fault-occurrence model captures The core Markov process, which could not be uniform in consistency. These formulas include protection patterns that show how immune the framework is to flaws.

Let $Q_w(z)$ stand for the likelihood that the number of covered defects will be in an operable state at time z. $\lambda_{j,j+1}$(rate of fault occurrence when the system is in state j) should be defined as the amount governs the transformation from state j to state j + 1. Let $q_j(x)$ show the likelihood distribution of the time needed to locate the error. As of now, we

$$Q_{j+1}(z) = \int_0^z Q_w(z-x)\lambda_{w,w+1}Q_j(x)\exp(-\lambda_{j+1,j+2}x)dx \quad (28)$$

With a simple fault administration mode, equation (19) yields the LT of qj(x). An Extended Stochastic Petri Net (ESPN) modeling framework is used in more complex scenarios. The next part provides a description of petri net models. Calculated integration can be used to assess equation (25).

To calculate the system's reliability R(z), Pr(z) is the likelihood that, at time z, exactly w faults were present, most of them have been addressed or the final issue is being attended to, and the system has not malfunctioned. We acquire by means of $Q_r(z)$.

$$P_{r+1}(z) = \int_0^z Q_r(x)\lambda_{j,j+1}p_f(x)\exp(-\lambda_{j+1,j+2}x)dx \quad (29)$$

The possibility that a single flaw won't cause a system failure in time x is indicated by pf(x), which is the counterpart of The malfunctioning mode of the whole thing. Now that you've noted the system's dependability.

$$R(z) = \sum_{j\varepsilon N} P_w(z) \quad (30)$$

where the operational state set N is shown. The value of pf(x) can also be obtained using the ESPN model.

4.2 Petri Net Models

Systems with concurrent, non-synchronous, or indeterminate behavior have been described using Petri nets. A Duality directed graph, also known as a Petri net, is composed of two types of nodes: a set of transitions T (represented by bars) and a collection of locations P (represented by circles); edges E connect the changes and locations. Switches can be made enabled by adding tokens to locations; a change is deemed enabled when a token appears in each of its input locations. Each submission site uses a single coin, and every output site yields one token per transaction location when the transition is activated. A Petri net's state (or marking) can be determined by tokens positioned at specific intervals, and state changes can be detected by transitions.

The fundamental Petri net model that was previously covered has a number of extensions. The likelihood curve (which adds non-determinism when allowing), the inhibitor curve, the OR logic, and the counter arc (which only permits a transition when the input arc contains more than k tokens) are a few variations.

It is not necessary to have tokens at every input location in order to activate a transition. ESPN employs the probability, counter-arc, and inhibitor models.

The core Petri net model that was previously covered has undergone a few changes. The counter curve (transition enabled only when the input arc includes at least k entries) and the probability arc (to provide non-determinism in engaging transitions), the OR logic, and the inhibitor arc are other examples. If tokens are not present in every input location, a transition can nevertheless be allowed. ESPN employs the probability, counter-arc, and inhibitor models.

Semi-Markov or time-homogeneous approaches can be used to analyze Petri net models thanks to Molloy's discovery that the marks a Markov process and a Petri net are compatible. When should the firing era start—when a transition is initiated or the first coin arrives is an issue for both stochastic Petri nets and timed transitions. It's crucial to take into account the likelihood that one period may end while another is still in progress. Resolving hiring-related disputes comes second. Likelihood is frequently allocated across the

stenciled area for models between the current and the next labeling that depend on a fixed firing time. Given the ejection rate, (which is determined by random discharge timings), stochastic Petri net models frequently anticipate an upcoming marking from the present one. There is an obstacle if some shifts are allowed to have no launching time. They are more likely to fire against one when such transitions are approved. Increasing fire rates throughout the changeover is the solution. Likelihood as determined at regular intervals in Petri nets. Similar to ESPN, this is accomplished by including inhibitor, probability, and counter curve.[22]

4.2 Dataflow Graph Models

Computer programmers and language designers have been actively examining the dataflow theory of computing in recent years. For computer system settings, dataflow graph models have shown to be excellent tools. Alternatively, the dataflow paradigm can be used to illustrate the unavoidable, delayed, or concurrent operation of machines. Data flowing graphs' tiny footprint, universal guidance, and clarity are their primary benefits over alternative models.

Furthermore, links and actors are the two types of nodes found in bipartite directed graphs, also known as dataflow graphs.

The edges that join the nodes create paths for communication. Actors are thought of as functions that are performed (as opposed to changes in Petri nets), while ties are regarded as token holders in place. It is possible to examine a dataflow graph model's dependability without taking into account the kinds of data tokens or the significance of the actions taken by operators. Tokens and links can be used as triggers by performers to act out a situation. We refer to some types of dataflow graphs as continuous dataflow graphs.

To put it simply, fresh coins are created on the resulting hyperlinks and coins are retrieved from the data links when an actor fires. Only when every input link on an actor has is the role eligible for execution if it has a token and each output connection has no extra tokens. Stated differently, new tokens are created on the output set and existing tokens on the input set are exhausted. By expanding the sequencing mode, actors can now be run when only a subset of output links—known as the output firing semantic set, F2—contain tokens and a subset of input links—known as the input striking conceptual set, F1) are empty. Non-determination may arise because shooting lexical sets can vary depending on how an actor performs in different situations.

One can illustrate the nondeterministic nature of implementation by providing estimations for the input and output shooting linguistic sets.[21]

Considering the firing circumstances, five different types of operators can be distinguished.

-Conjunctive operator: For the character to become active, coins must be present in each input connection.

-Disjunctive operator: The participant cannot be terminated unless an element is found on one of the input links.

-Collective operator: In order for the participant to be activated, coins need to be present in at least one of the input connections. This paper does not take collective participants into consideration.

-Selective operator: Only one of the output connections gets an identification code performer flames.

-Distributive operator: When the trigger flames, coins get transmitted to each downstream link.

Figure 3 provides a graphic depiction of the various actor classes. The path that connects actor ai to actor aj (which does not include actor aj) is represented as Rij, Cj indicates the dependability of connection j, or the communication channel, and Rat provides the actor's dependability.

The reliability of a dataflow graph can be evaluated by examining the likelihood that a series of tasks assigned to the graph's actors will be completed successfully. If this sequence is discovered to reflect a particle's path across the graph, then dependability is the chance that the particle will follow a successful path.

Since dataflow graphs are hierarchical networks, there are two methods to assess the reliability of a the data flow graph. First, the reliability of the subsections is determined. In the subsequent phase, the sub-graphs' reliabilities are accurately integrated by utilizing the chart's hierarchical composition. HARP and this deconstruction and aggregation process are comparable.

Conversely, dataflow graph models offer structural as well as behavioral dissection. Functional interpretation of dataflow graph models is an additional option.

Petri nets, on the other hand, are limited to continuous use.

The equations Figure 3 should be utilized to take look at all sorts of dataflow operators. When integrating portions reliabilities. This paper does not take collective actors into consideration. The following details set dataflow graph pathways apart from parallel paths and dependability network series. (A) Participants that are conjunctional and distribute point in separate directions, both of which is required for proper operation.. When the paths are independent of one other, the dependability of the parallel path graph is calculated as the sum of the accuracy of the individual pathways. (a) Selective and disjunctive agents produce multiple pathways, but only one is required. By combining the likelihoods of the various paths and applying weights based on path likelihood, the dependability of the graph featuring these routes is determined. (c) When the pathways are not standalone, a reliant structure specifies the manner in which the dependability's of several participants (or portions) are combined to illustrate the hierarchy's sturdiness.

To assess a dataflow graph's dependability, take the next few steps.

(1) List the paragraphs' subheadings.

(2) Use the recursive version of this approach or alternative techniques, such as Markov chains, HARP, or CARE III, to determine the reliabilities of sub graphs.

(3) Create a simplified graph by replacing sub graphs with single actors.
(4) Identify unique paths.
(5) Utilizing actor types, integrate performer (sub graph) reliabilities to determine each path's dependability. Combine the path reliabilities to get the graph's overall faithfulness.
Employed Petri nets both the math and logic functions in a single accumulator to replicate the logic of control flow during the operation of a command. The dataflow equivalent of a Petri net is shown in Figure 4.[19]

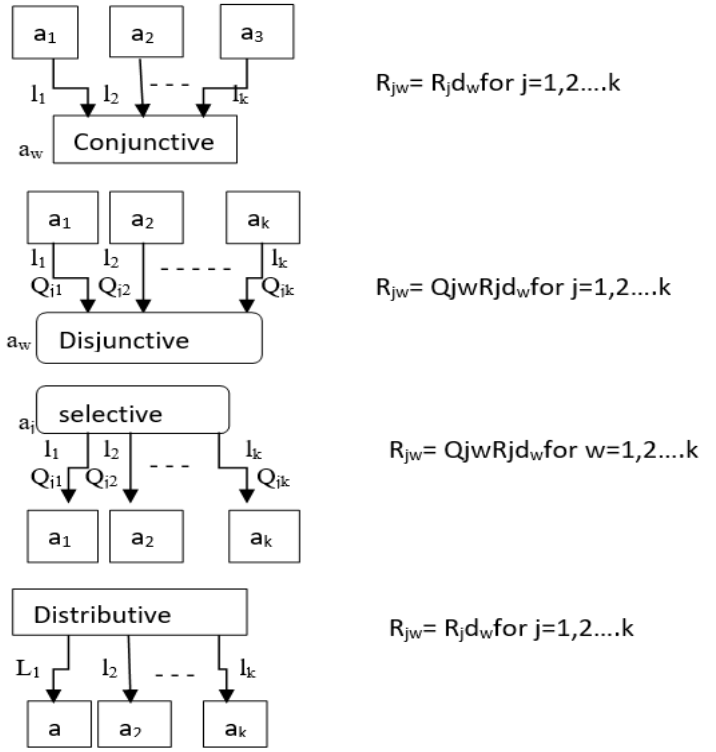
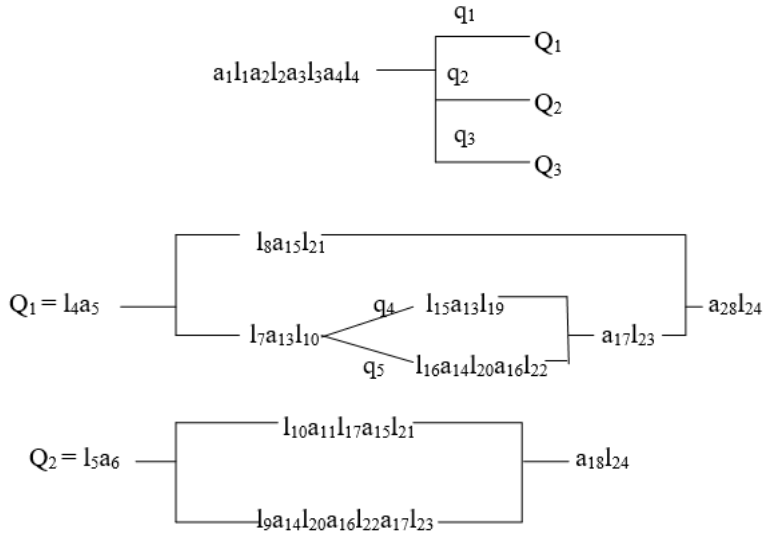


Figure 3 Three ways to express dataflow participants' reliability.

In order to make interpretation easier, the actors are purposefully given names by the occurrences.
In figure 4, the three unique pathways Q_1, Q_2, and Q_3 are shown
The frequency of Conditional is shown by the probabilityq_1 ,q_2 and q_3. In a typical program, the store and arithmetic instructions correspond to the q_4 and q_5 probability that a condition will be met or notRemember that the output firing semantic sets were used to determine these probabilities. Semantic set firing probabilities are significant exclusively to collective agents.



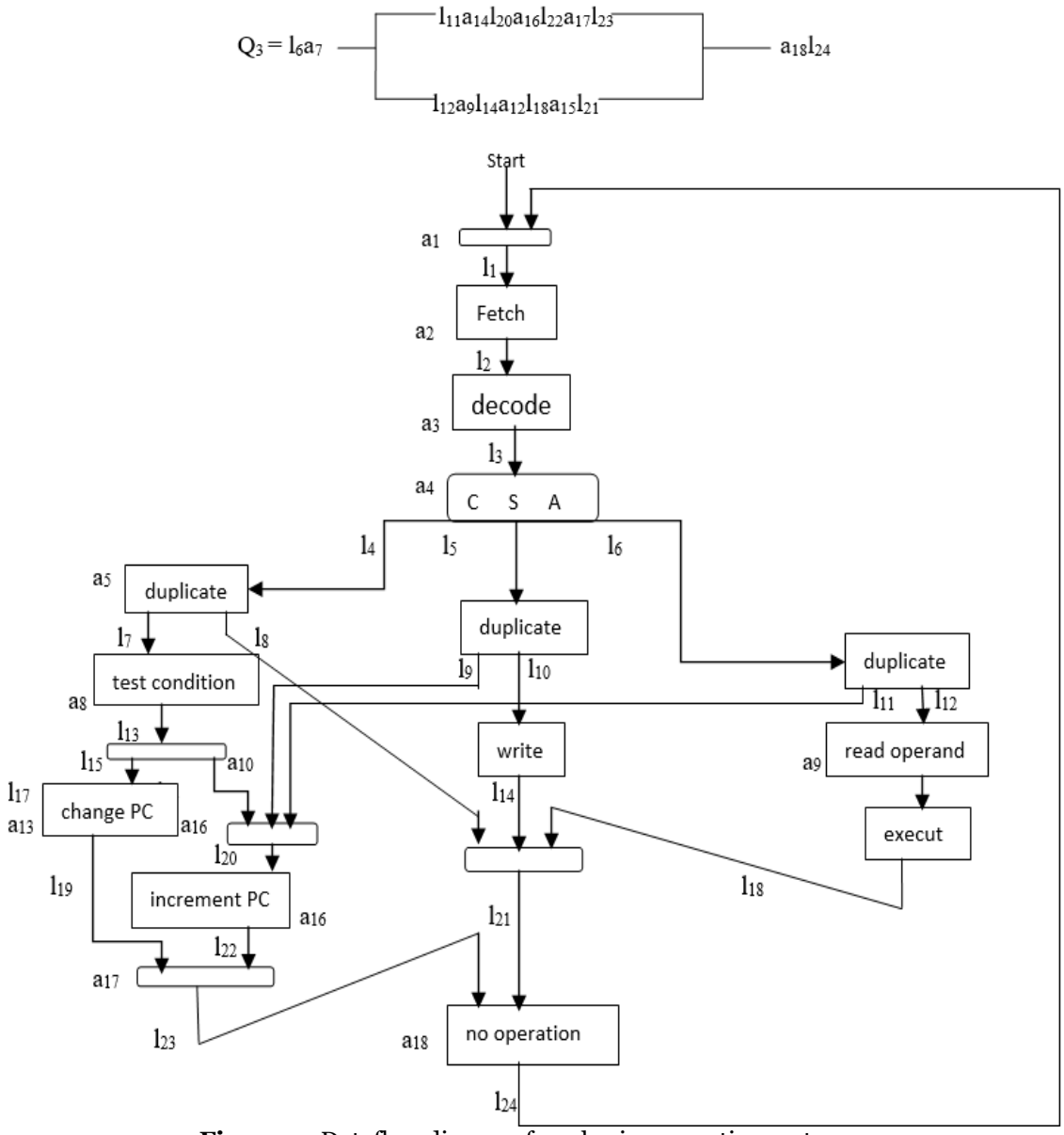


Figure 4. Dataflow diagram for a basic computing system.

Path Q_l 's trustworthiness is indicated by

$$R^{(1)} = C_4R_5C_8R_{15}C_{21}C_7R_8C_{13}R_{10}(p_4C_{15}R_{13}C_{19}+q_5C_{16}R_{14}C_{20}R_{16}C_{22}) *R_{17}C_{23}R_{18}C_{24}$$

In a similar way, trustworthiness $R^{(2)}$ and $R^{(3)}$of pathwaysQ_2 and Q_3can be found. Next, the system's dependability is provided by

$$R(T) = R_1C_1R_2C_2R_3C_3(q_1R^{(1)} + q_2R^{(2)})$$

Here, dataflow models are used to calculate a bridge network's reliability. A bridge network is depicted in Figure 5a, and the dataflow Figure displays the network's graph model. 5b. Two distinct actors (a_5 , a_6) are used to illustrate the bidirectional character of unit E for the sake of depiction. The dataflow graph in figure 5b shows four different pathways.[21]

Q_1: $l_0a_0l_1a_1l_3a_3l_5a_7l_{11}a_9l_{13}a_{11}l_{15}$
Q_2: $l_0a_0l_1a_1l_3a_3l_6a_5l_9a_8l_{12}a_{10}l_{14}a_{11}l_{15}$
Q_3: $l_0a_0l_2a_2l_4a_4l_8a_8l_{12}a_{10}l_{14}a_{11}l_{15}$
Q_4: $l_0a_0l_2a_2l_4a_4l_7a_6l_{10}a_7l_{11}a_9l_{13}a_{11}l_{15}$

The four pathways are not autonomous, though. Assumed is the following dependence structure.
Path Q_3 is utilized when every unit is operational; Q_1 is only used in the event that unit C is inoperative functioning or when unit C is operating D is not, however; While units A and D are not, functioning, Q_4 is utilized; If units B and C don't match functioning, $Q_2$is utilized.

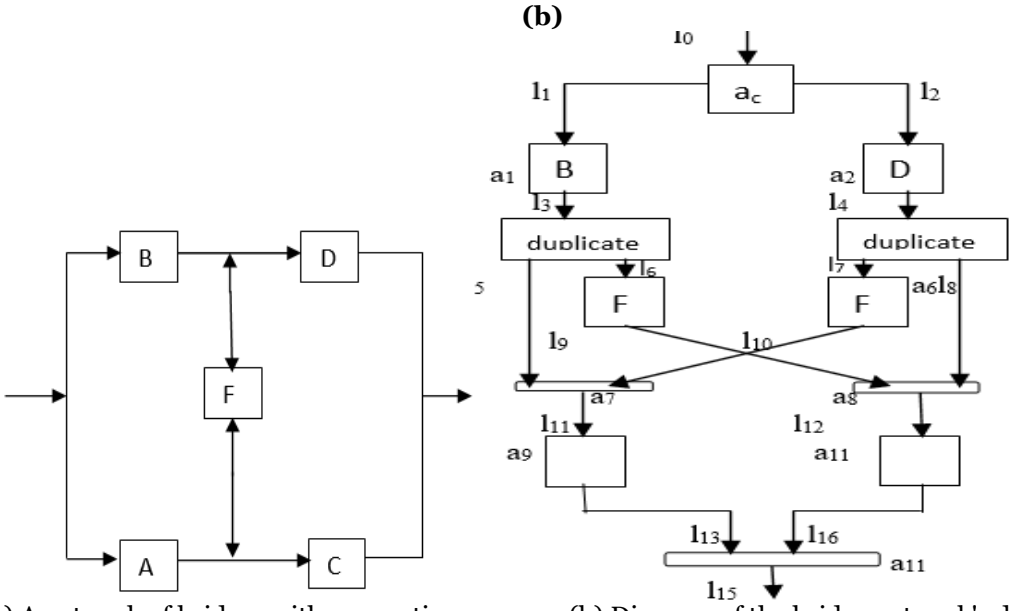


Figure5. (a) A network of bridges with connections. (b) Diagram of the bridge network's data flow

The veracity of every actor, with the exception of a_1 (unit B), a_9 (unit A), a_2 (unit D), is provided for convenience. a_5. a_6 (unit F), a_{10} (unit C). are configured to 1. Additionally, all link dependability are set to 1. One may compute the dependability the bridge network is identifiable in the the data flow graph as

$$R(\text{Bridge Network}) = R^{(3)} + R^{(1)}[(1-R_D)+R_D(1-R_C)]+R^{(2)}(1-R_A)R_D(1-R_D)+R^{(4)}(1-R_B)R_D(1-R_C)$$
$$=R_DR_C+R_BR_A[(1-R_D)+R_D(1-R_C)]+R_BR_FR_C(1-R_A)R_D(1-R_D)+R_DR_FR_B(1-R_B)R_D(1-R_C)$$

Where path Q_j dependability is denoted by $R^{(1)}$. Units A, B, C, D, and F have reliabilities denoted as R_B, R_A, R_D, R_C, and R_F, in that order.

5 Conclusion

This overview's goal has been to list the many modeling approaches that a reliability scientist who works with computer systems can use. Structure models alone are insufficient since the time the aspect is so significant to the assessment of reliability. The method that combines a stochastic process model with the unique features of the structure appears to be the most effective. Because stochastic models can accurately represent and analyze the uncertainties, variability, and randomness present in complex systems, they are frequently seen as beneficial in the field of computing. Stochastic models give useful insights for controlling and optimizing system reliability in dynamic and unpredictable contexts by offering a flexible framework, permitting risk assessment, and facilitating performance evaluation through simulation. Although they are not always suitable, their ability to manage probabilistic components makes them a better option in situations when accurate portrayal of uncertainty is essential. This method has led to the introduction of other models (CARE 111, HARP, Dataflow, etc.). These methods are relatively new, and numerous government, business, and academic entities are now conducting more research on them. Time is not a continuous parameter in the dataflow models shown here.

Similar to Petri net marks, an unbroken dataflow graph's state can be determined. Dataflow graph models can then be exposed to both discrete and continuous Markov analysis techniques by defining Markov processes with the markings. The ultimate choice will surely depend on the trade-off between the model's tractability and degree of realism, since mathematical modeling is the process of simulating the functioning of a real system.

Reference

1. Kayid, M.; Alshehri, M.A. Stochastic Comparisons of Lifetimes of Used Standby Systems. Mathematics 2023, 11, 3042.
2. Shrahili, M. and Kayid, M. (2023) 'Stochastic orderings of the idle time of inactive standby systems', *Mathematics*, 11(20), p. 4303. doi:10.3390/math11204303.
3. Yadav, R.K. and Malik, S.C. (2020) 'Stochastic analysis of a computer system with unit wise cold standby redundancy and failure of service facility', *International Journal of Mathematical, Engineering and Management Sciences*, 5(3), pp. 529–543. doi:10.33889/ijmems.2020.5.3.044.

4. Mishra, A., Jagannatham, A.K. and Hanzo, L. (2020) 'Sparse bayesian learning-aided joint sparse channel estimation and ML sequence detection in space-time trellis coded MIMO-OFDM Systems', *IEEE Transactions on Communications*, 68(2), pp. 1132–1145. doi:10.1109/tcomm.2019.2953260.
5. Navarro, J.; Calì, C. Inactivity times of coherent systems with dependent components under periodical inspections. Appl. Stoch.Model. Bus. Ind. 2019, 35, 871–892.
6. Salehi, E.; Tavangar, M. Stochastic comparisons on conditional residual lifetime and inactivity time of coherent systems with exchangeable components. Stat. Probab. Lett. 2019, 145, 327–337.
7. Zhu, M. and Pham, H. (2019) 'A novel system reliability modeling of hardware, software, and interactions of hardware and software', *Mathematics*, 7(11), p. 1049. doi:10.3390/math7111049.
8. Eryilmaz, S.; Tekin, M. Reliability evaluation of a system under a mixed shock model. J. Comput. Appl. Math.2019, 352, 255–261.
9. Gao, X.;Wang, R.; Gao, J.; Gao, Z.; Deng,W. A novel framework for the reliability modelling of repairable multistate complex mechanical systems considering propagation relationships. Qual. Reliab. Eng. Int. 2019, 35, 84–98.
10. Li, G.; Zhu, H.; He, J.;Wu, K.; Jia, Y. Application of power law model in reliability evaluation of machine tools by considering working condition di_erence. Qual. Reliab. Eng. Int. 2019, 35, 136–145.
11. Jia, G.; Gardoni, P. State-Dependent stochastic models: A general stochastic framework for modeling deteriorating engineering systems considering multiple deterioration process and their interactions. Struct.Saf. 2018, 72, 99–110.
12. Zhu, M.; Pham, H. A two-phase software reliability modeling involving with software fault dependency and imperfect fault removal. Comput. Lang. Syst. Struct. 2018, 53, 27–42.
13. Zhu, M.; Pham, H. A software reliability model incorporating martingale process with gamma-distributed environmental factors. Ann. Oper. Res. 2018, 1–22.
14. Rodríguez-Borbón, M.I.; Rodríguez-Medina, M.A.; Rodríguez-Picón, L.A.; Alvarado-Iniesta, A.; Sha, N.Reliability estimation for accelerated life tests based on a Cox proportional hazard model with error e_ect.Qual. Reliab. Eng. Int. 2017, 33, 1407–1416.
15. Yi, X.J.; Shi, J.; Dhillon, B.S.; Hou, P.; Lai, Y.H. A new reliability analysis method for repairable systems with multifunction modes based on goal-oriented methodology. Qual. Reliab. Eng. Int. 2017, 33, 2215–2237.
16. Lung, C.H.; Zhang, X.; Rajeswaran, P. Improving software performance and reliability in a distributed and concurrent environment with an architecture-based self-adaptive framework. J. Syst. Softw. 2016, 121,311–328.
17. Park, J.; Baik, J. Improving software reliability prediction through multi-criteria based dynamic model selection and combination. J. Syst. Softw. 2015, 101, 236–244.
18. Wang, S.;Wu, Y.; Lu, M.; Li, H. Discrete nonhomogeneous Poisson process software reliability growth models based on test coverage. Qual. Reliab. Eng. Int. 2013, 29, 103–112.
19. Yang, Q.; Zhang, N.; Hong, Y. Reliability analysis of repairable systems with dependent component failures under partially perfect repair. IEEE Trans. Reliab. 2013, 62, 490–498.
20. Amiripour, F.; Khaledi, B.E.; Shaked, M. Stochastic orderings of convolution residuals. Metrika2013, 76, 559–576.
21. Arnold, B.C.; Villasenor, J.A. Exponential characterizations motivated by the structure of order statistics in samples of size two.Stat. Probab. Lett. 2013, 83, 596–601
22. Kavi, K.M. and Bhat, U.N. 'Reliability Analysis of computer systems using Dataflow Graph Models', *IEEE Transactions on Reliability*, 35(5), pp. 529–531. doi:10.1109/tr.1986.4335538.
23. Bhat, U.N. and Kavi, K.M. 'Reliability models for computer systems: An overview including dataflow graphs', *Sadhana*, 11(1–2), pp. 167–186. doi:10.1007/bf02811317.