Volume 3, Issue 6, June 2013 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Estimating Reliability of Web Application Using Markov Model

Divya Bindal
Department of Computer Science and Engineering
University Institute of Engineering & Technology
Kurukshetra University,Kurukshetra,
Haryana, India

Abstract— Software reliability is the most important aspect of any software. To improve the performance of software assessment of reliability is required. A Markov Model is a mathematical system consisting of finite no of possible states. It undergoes transition from one state to another state. The Markov Model assumes Markov property which states that that future is independent of the past i.e. the probability of transitioning from some state i to another state j depends only on the current state i not on the sequence of events that preceded it . Markov Model is used to represent the architecture of the software & provides a mean for analyzing the reliability of software. In this paper we have developed Markov Model of a web application used in the real world. It consists of 6 states and 30 transitions then derive matrix for it after that derived matrix is used for estimating reliability of each state of web application.

Keywords— Markov Model, Reliability, Failure Rate, Statistical Testing, Probability

I. INTRODUCTION TO MARKOV MODEL

Software reliability is the possibility that user will use software without encountering its failure in actual usage environment. Over the past two decades, a number of studies have been conducted for measuring reliability of given software. As a result, a number of analytical models have been introduced. Most of these analytical models focus on observing the behaviours of software, based on an operational profile. Measurements are made using the data collected over the observation period, regardless of the structure of the software. They are mainly applied during the late phase of the software development in order to determine if the software meets its reliability requirements. These models are difficult to apply if no sufficient test data is available or when changes are made to the software. Other utilizes Markov property and makes use of the software architecture, which is defined as Markov Model “A Markov Model is a mathematical system that undergoes transitions from one state to another, between a finite or countable number of possible states. It is a random process usually characterized as memory less: the next state depends only on the current state and not on the sequence of events that preceded it” [1]. By using Markov Model a statistical model of software is drawn wherein each possible use of the software has an associated probability of occurrence. Test cases are drawn from the sample population of possible uses according to the sample distribution and run against the software under test. Various statistics of interest, such as the estimated failure rate and mean time to failure of the software are computed.

A sample two state Markov Model

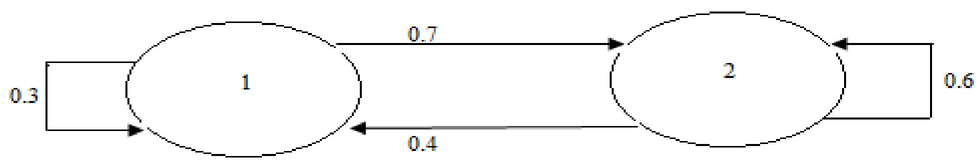


Figure1 Markov Model

A usage model for a software system consists of states, i.e., externally visible modes of operation that must be maintained in order to predict the application of all system inputs, and state transitions that are labelled with system inputs and transition probabilities. To determine the state set, one must consider each input and the information necessary to apply that input. It may be that certain software modes cause an input to become more or less probable (or even illegal). Such a mode represents a state or set of states **in** the usage chain. Once the states are identified, we establish a start state, a terminate state, and draw a state transition diagram by considering the effect of each input from each of the identified states. The Markov chain is completely defined when transition probabilities are established that represent the best estimate of real usage.

Advantages of using Markov Model

- **Simplistic Modeling Approach:** Markov Models are simple to generate
- **Redundancy Management Techniques**: System reconfiguration required by failures is easily incorporated in the model.
- **Coverage:** Covered and uncovered failures of components are mutually exclusive events.
- **Complex Systems:** Many simplifying techniques exit which allow the modeling of complex systems.
- **Sequenced Events:** It allow computing of an event resulting from a sequence of sub events.

Disadvantage of Markov Model

The major drawback of Markov Model is the explosion of number of states as the size of system increases. The resulting models are large & complicated.

II. TYPES OF MARKOV MODEL

The most common Markov models and their relationships are summarized in the following table1 [3]:

Table 1 Types of Markov Model

	System state is fully observable	**System state is partially observable**
System is autonomous	Markov Chain	Hidden Markov Model
System is controlled	Markov Decision Process	Partially observable Markov Decision Process

- **Markov Chain:** The simplest Markov model is the Markov chain. It models the state of a system with a random variable that changes through time. In this context, the Markov property suggests that the distribution for this variable depends only on the distribution of the previous state.

- **Hidden Markov Model:** A hidden Markov model is a Markov chain for which the state is only partially observable. In other words, observations are related to the state of the system, but they are typically insufficient to precisely determine the state.

- **Markov Decision Process:** A Markov decision process is a Markov chain in which state transitions depend on the current state and an action vector that is applied to the system. Typically, a Markov decision process is used to compute a policy of actions that will maximize some utility with respect to expected rewards.

- **Partially Observable Markov Decision Process:** A partially observable Markov decision process (POMDP) is a Markov decision process in which the state of the system is only partially observed. POMDPs are known to be NP complete, but recent approximation techniques have made them useful for a variety of applications, such as controlling simple agents or robots.

III. SOFTWARE RELIABILITY

Software Reliability is defined as the ability of software to function under given environmental conditions for particular amount of time by taking into account all precisions of the software [4].

Software reliability testing:

Software reliability testing is one of the testing field, which deals with checking the ability of software to function under given environmental conditions for particular amount of time by taking into account all precisions of the software. In Software Reliability Testing, the problems are discovered regarding the software design and functionality and the assurance is given that the system meets all requirements. Software Reliability is the probability that software will work properly in specified environment and for given time.

Probability = Number of cases when we find failure / Total number of cases under consideration

© 2013, IJARCSSE All Rights Reserved

Objective of reliability testing:

The main objective of the reliability testing is to test the performance of the software under given conditions without any type of corrective measure with known fixed procedures considering its specifications.

Need of Reliability Testing:

Nowadays in maximum number of fields we find the application of computer software. Also some software is used in many critical applications like in industries, in military, in commercial systems etc. For this software from last century software engineering is developing. There is no complete measure to assess them. But to assess them software reliability measure are used as tool. So software reliability is the most important aspect of any software. To improve the performance of software product and software development process through assessment of reliability is required. Reliability testing is of great use for software managers and software practitioners. Thus ultimately testing reliability of software is important.

IV. TESTING PROCESS

The typical process followed when performing statistical testing is presented in Figure2.

- **Get Specification:** Some form of a specification detailing the correct behaviour of the software is needed to develop the usage model. The correct behaviour of the software may be defined by a formal specification, requirements documentation, user's manual, or a predecessor system.
- **Develop Model Structure:** The states of use and arcs connecting the states are identified. In current practice this stage is manual, i.e., the structure of the usage model cannot be derived automatically from a specification or other artefact. However, work is in progress to show how to derive the model structure from a sequence-based specification of the software
- **Assign Probabilities:** The state transition probabilities of the usage model are assigned. The probabilities may be assigned manually or they may be represented as a set of constraints and automatically calculated to satisfy some testing goal.

Testing Process

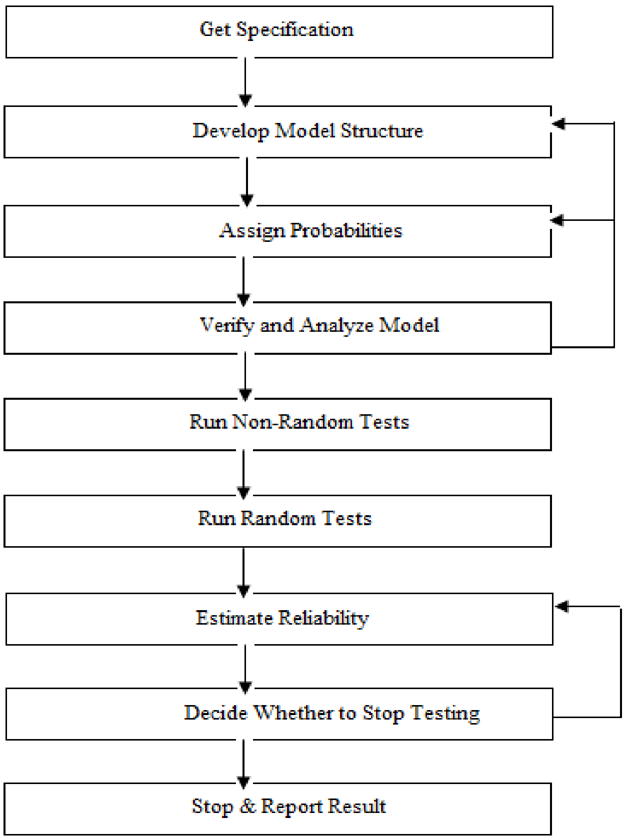


Figure2 Testing Process

© 2013, IJARCSSE All Rights Reserved

- **Verify and Analyze Model:** Analytical results are calculated to aid in test planning and verifying that the model properly represents the expected use of the software
- **Run Non-Random Tests:** Non-random test cases are crafted or generated from the usage model and then executed on the software under test. Examples of non-random tests currently being used are hand crafted tests, test cases generated in order of probability of occurrence, and test cases generated to cover all arcs in the usage model in the minimum number of testing steps. Non-random tests may be run to satisfy contractual obligations, explore a particular use of the software, help validate the usage model and test facility, or determine whether the software is stable enough for full scale testing. Current practice does not include use of non-random tests in reliability estimation.
- **Estimate Reliability:** The testing record containing information as to which tests were run and where failures were observed in the test cases is used to estimate the reliability of the software.
- **Decide Whether to Stop Testing**: The testing record is evaluated to determine whether testing should continue or stop.
- **Stop and Report Results:** After testing is finished the test results may be used for a number of purposes, such as deciding whether to release the product, evaluating whether the software development process is under control, or evaluating the performance of a new piece of technology used in the product.

V. MARKOV MODEL OF A WEB APPLICATION

We have developed Markov Model of a Web application software used in the real world. It consists of 6 states and 30 transitions.

- **State1** named as "P1" having five transactions [λ_{11}, λ_{12}, λ_{13}, λ_{14}, λ_{15}].
- **State2** named as "P2" is not actually exist it represent three other states P2(A),P2(B),P2(C) these states are not connected with each other & Both having five transactions [λ_{21}, λ_{22}, λ_{23}, λ_{24}, λ_{25}].
  These states are represented as:

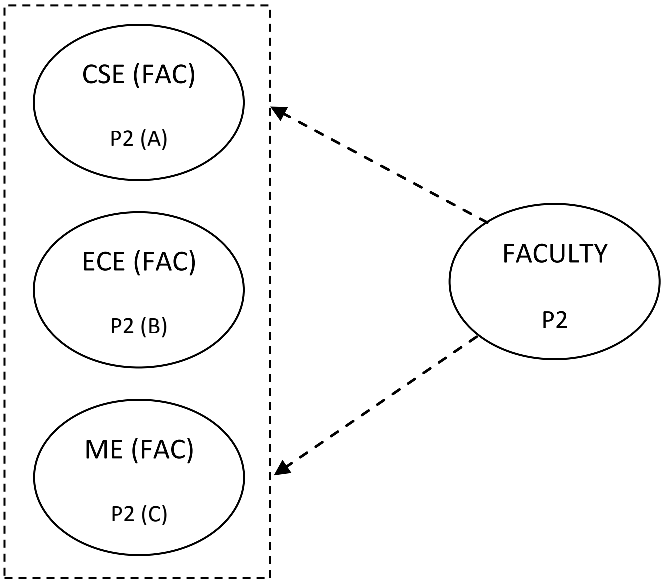


Figure3

- **State3** named as "P3" having five transactions [λ_{31}, λ_{32}, λ_{33}, λ_{34}, λ_{35}].
- **State4** named as "P4" & also having five transactions [λ_{41}, λ_{42}, λ_{43}, λ_{44}, λ_{45}].
- **State5** named as "P5" is not actually exist it represent three other states $P5_1$, $P5_2$, $P5_3$ .These states are connected with each other & Both having five common Transactions [λ_{51}, λ_{52}, λ_{53}, λ_{54}, λ_{55}] & two Diff transactions. These states are represented as Figure 5(b) drawn below.
- **State6** named as "P6" it is an exit state & having no transaction.

© 2013, IJARCSSE All Rights Reserved

Markov Model of a web application:

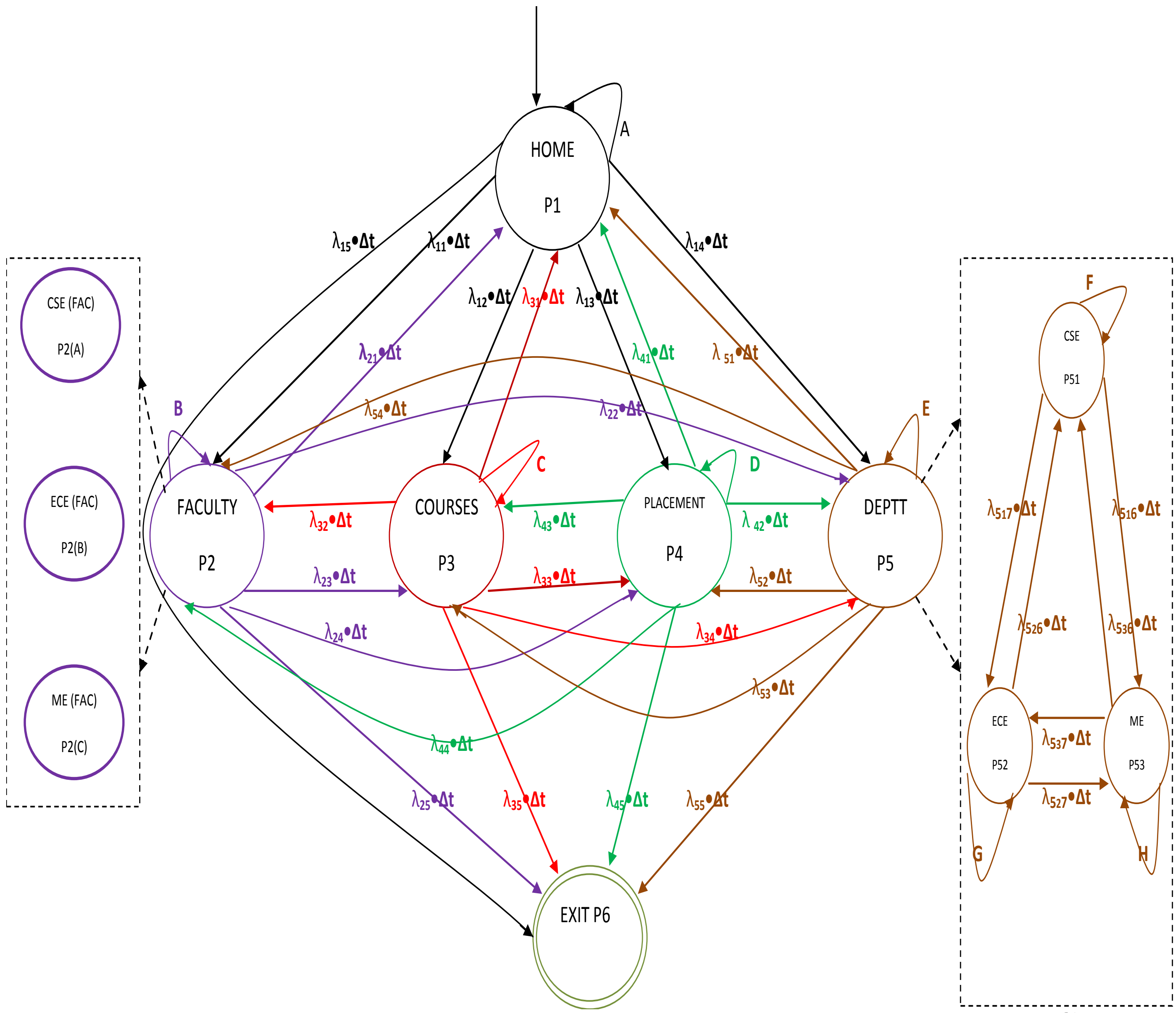


Figure4 (b)

Figure4 (a) Markov Model of a college website

TABLE 2 Symbols Used in above Figure

S.No	SYMBOL	DESCRIPTION
1	λ	Failure Rate
2	A	[1-(λ11 + λ12 + λ13 + λ14 + λ15)Δt]
3	B	[1-(λ21 + λ22 + λ23 + λ24 + λ25)Δt]
4	C	[1-(λ31 + λ32 + λ33 + λ34 + λ35)Δt]
5	D	[1-(λ41 + λ42 + λ43 + λ44 + λ45)Δt]
6	E	[1-(λ51 + λ52 + λ53 + λ54 + λ55)Δt]
10	F	[1-(λ51 + λ52 + λ53 + λ54 + λ55 + λ516 + λ517)Δt]
11	G	[1-(λ51 + λ52 + λ53 + λ54 + λ55 + λ526 + λ527)Δt]
12	H	[1-(λ51 + λ52 + λ53 + λ54 + λ55 + λ536 + λ537)Δt]

© 2013, IJARCSSE All Rights Reserved

Probability of being in each of the states at time t+Δt:

$$P1(t+\Delta t) = [[P1(t)\bullet\{1-(\lambda_{11}+\lambda_{12}+\lambda_{13}+\lambda_{14}+\lambda_{15})\Delta t)]+[P2(t)\bullet\lambda_{21}\bullet\Delta t]+[P3(t)\bullet\lambda_{31}\bullet\Delta t] +[P4(t)\bullet\lambda_{41}\bullet\Delta t]+[P5(t)\bullet\lambda_{51}\bullet\Delta t]]$$

$$P2(t+\Delta t) = [[P2(t)\bullet\{1-(\lambda_{21}+\lambda_{22}+\lambda_{23}+\lambda_{24}+\lambda_{25})\Delta t\}]+[P1(t)\bullet\lambda_{11}\bullet\Delta t]+[P3(t)\bullet\lambda_{32}\bullet\Delta t] +[P4(t)\bullet\lambda_{44}\bullet\Delta t]+[P5(t)\bullet\lambda_{54}\bullet\Delta t]]$$

$$P3(t+\Delta t) = [[P3(t)\bullet\{1-(\lambda_{31}+\lambda_{32}+\lambda_{33}+\lambda_{34}+\lambda_{35})\Delta t\}]+[P1(t)\bullet\lambda_{12}\bullet\Delta t]+[P2(t)\bullet\lambda_{23}\bullet\Delta t] +[P4(t)\bullet\lambda_{43}\bullet\Delta t]+[P5(t)\bullet\lambda_{53}\bullet\Delta t]]$$

$$P4(t+\Delta t) = [[P4(t)\bullet\{1-(\lambda_{41}+\lambda_{42}+\lambda_{43}+\lambda_{44}+\lambda_{45})\Delta t\}]+[P1(t)\bullet\lambda_{13}\bullet\Delta t]+[P2(t)\bullet\lambda_{24}\bullet\Delta t] +[P3(t)\bullet\lambda_{33}\bullet\Delta t]+[P5(t)\bullet\lambda_{52}\bullet\Delta t]]$$

$$P5(t+\Delta t) = [[P5(t)\bullet\{1-(\lambda_{51}+\lambda_{52}+\lambda_{53}+\lambda_{54}+\lambda_{55})\Delta t\}]+[P1(t)\bullet\lambda_{14}\bullet\Delta t]+[P2(t)\bullet\lambda_{22}\bullet\Delta t] +[P3(t)\bullet\lambda_{34}\bullet\Delta t]+[P4(t)\bullet\lambda_{42}\bullet\Delta t]]$$

$$P5_1(t+\Delta t) = [[P5_1(t)\bullet\{1-(\lambda_{51}+\lambda_{52}+\lambda_{53}+\lambda_{54}+\lambda_{55}+\lambda_{5_16}+\lambda_{5_17})\Delta t\}]+[P1(t)\bullet\lambda_{14}\bullet\Delta t] +[P2(t)\bullet\lambda_{22}\bullet\Delta t]+[P3(t)\bullet\lambda_{34}\bullet\Delta t]+[P4(t)\bullet\lambda_{42}\bullet\Delta t]+[P5_2(t)\bullet\lambda_{5_26}\bullet\Delta t]+[P5_3(t)\bullet\lambda_{5_36}\bullet\Delta t]]$$

$$P5_2(t+\Delta t) = [[P5_2(t)\bullet\{1-(\lambda_{51}+\lambda_{52}+\lambda_{53}+\lambda_{54}+\lambda_{55}+\lambda_{5_26}+\lambda_{5_27})\Delta t\}]+[P1(t)\bullet\lambda_{14}\bullet\Delta t] +[P2(t)\bullet\lambda_{22}\bullet\Delta t]+[P3(t)\bullet\lambda_{34}\bullet\Delta t]+[P4(t)\bullet\lambda_{42}\bullet\Delta t]+[P5_1(t)\bullet\lambda_{5_17}\bullet\Delta t]+[P5_3(t)\bullet\lambda_{5_37}\bullet\Delta t]]$$

$$P5_3(t+\Delta t) = [[P5_3(t)\bullet\{1-(\lambda_{51}+\lambda_{52}+\lambda_{53}+\lambda_{54}+\lambda_{55}+\lambda_{5_36}+\lambda_{5_37})\Delta t\}]+[P1(t)\bullet\lambda_{14}\bullet\Delta t] +[P2(t)\bullet\lambda_{22}\bullet\Delta t]+[P3(t)\bullet\lambda_{34}\bullet\Delta t]+[P4(t)\bullet\lambda_{42}\bullet\Delta t]+[P5_1(t)\bullet\lambda_{5_16}\bullet\Delta t]+[P5_2(t)\bullet\lambda_{5_27}\bullet\Delta t]]$$

$$P6(t+\Delta t) = [[[P9(t)\bullet\Delta t]+[P1(t)\bullet\lambda_{15}\bullet\Delta t]+[P2(t)\bullet\lambda_{25}\bullet\Delta t]+[P3(t)\bullet\lambda_{35}\bullet\Delta t] +[P4(t)\bullet\lambda_{45}\bullet\Delta t]+[P5(t)\bullet\lambda_{55}\bullet\Delta t]+]$$

Rearranging the above equations:

$$[P1(t+\Delta t)-P1(t)]/\Delta t = [[-(\lambda_{11}+\lambda_{12}+\lambda_{13}+\lambda_{14}+\lambda_{15})P1(t)]+[P2(t)\bullet\lambda_{21}]+[P3(t)\bullet\lambda_{31}]+[P4(t)\bullet\lambda_{41}] +[P5(t)\bullet\lambda_{51}]]$$

$$[P2(t+\Delta t)-P2(t)]/\Delta t = [[-(\lambda_{21}+\lambda_{22}+\lambda_{23}+\lambda_{24}+\lambda_{25})P2(t)]+[P1(t)\bullet\lambda_{11}]+[P3(t)\bullet\lambda_{32}]+[P4(t)\bullet\lambda_{44}(t)] +[P5(t)\bullet\lambda_{54}]]$$

$$[P3(t+\Delta t)-P3(t)]/\Delta t = [[-(\lambda_{31}+\lambda_{32}+\lambda_{33}+\lambda_{34}+\lambda_{35})P3(t)]+[P1(t)\bullet\lambda_{12}]+[P2(t)\bullet\lambda_{23}]+[P4(t)\bullet\lambda_{43}] +[P5(t)\bullet\lambda_{53}]]$$

$$[P4(t+\Delta t)-P4(t)]/\Delta t = [[-(\lambda_{41}+\lambda_{42}+\lambda_{43}+\lambda_{44}+\lambda_{45})P4(t)\}]+[P1(t)\bullet\lambda_{13}]+[P2(t)\bullet\lambda_{24}] +[P3(t)\bullet\lambda_{33}]+[P5(t)\bullet\lambda_{52}]]$$

$$[P5(t+\Delta t)-P5(t)]/\Delta t = [[-(\lambda_{51}+\lambda_{52}+\lambda_{53}+\lambda_{54}+\lambda_{55})P5(t)\}]+[P1(t)\bullet\lambda_{14}]+[P2(t)\bullet\lambda_{22}]+[P3(t)\bullet\lambda_{34}] +[P4(t)\bullet\lambda_{42}]]$$

$$[P5_1(t+\Delta t)-P5_1(t)]/\Delta t=[[-(\lambda_{51}+\lambda_{52}+\lambda_{53}+\lambda_{54}+\lambda_{55}+\lambda_{5_16}+\lambda_{5_17})\Delta t\}]+[P1(t)\bullet\lambda_{14}]+[P2(t)\bullet\lambda_{22}]$$
$$+[P3(t)\bullet\lambda_{34}]+[P4(t)\bullet\lambda_{42}]+[P5_2(t)\bullet\lambda_{5_26}]+[P5_3(t)\bullet\lambda_{5_36}]]$$

$$[P5_2(t+\Delta t)-P5_2(t)]/\Delta t=[[-(\lambda_{51}+\lambda_{52}+\lambda_{53}+\lambda_{54}+\lambda_{55}+\lambda_{5_26}+\lambda_{5_27})P5_2(t)]+[P1(t)\bullet\lambda_{14}]$$
$$+[P2(t)\bullet\lambda_{22}]+[P3(t)\bullet\lambda_{34}]+[P4(t)\bullet\lambda_{42}]+[P5_1(t)\bullet\lambda_{5_17}]+[P5_3(t)\bullet\lambda_{5_37}]]$$

$$[P5_3(t+\Delta t)-P5_3(t)]/\Delta t=[[-(\lambda_{51}+\lambda_{52}+\lambda_{53}+\lambda_{54}+\lambda_{55}+\lambda_{5_36}+\lambda_{5_37})P5_3(t)]+[P1(t)\bullet\lambda_{14}]$$
$$+[P2(t)\bullet\lambda_{22}]+[P3(t)\bullet\lambda_{34}]+[P4(t)\bullet\lambda_{42}]+[P5_1(t)\bullet\lambda_{5_16}]+[P5_2(t)\bullet\lambda_{5_27}]]$$

$$[P6(t+\Delta t)-P9(t)]/\Delta t=[P1(t)\bullet\lambda_{15}]+[P2(t)\bullet\lambda_{25}]+[P3(t)\bullet\lambda_{35}]+[P4(t)\bullet\lambda_{45}]$$
$$+P5(t)\bullet\lambda_{55}]]$$

Taking the limit as $\Delta t \rightarrow 0$:

$$dP1(t)/dt=[[-(\lambda_{11}+\lambda_{12}+\lambda_{13}+\lambda_{14}+\lambda_{15})P1(t)]+[P2(t)\bullet\lambda_{21}]+[P3(t)\bullet\lambda_{31}]+[P4(t)\bullet\lambda_{41}]+[P5(t)\bullet\lambda_{51}]]$$

$$dP2(t)/dt=[[-(\lambda_{21}+\lambda_{22}+\lambda_{23}+\lambda_{24}+\lambda_{25})P2(t)]+[P1(t)\bullet\lambda_{11}]+[P3(t)\bullet\lambda_{32}]+[P4(t)\bullet\lambda_{44}(t)]+[P5(t)\bullet\lambda_{54}]]$$

$$dP3(t)/dt=[[-(\lambda_{31}+\lambda_{32}+\lambda_{33}+\lambda_{34}+\lambda_{35})P3(t)]+[P1(t)\bullet\lambda_{12}]+[P2(t)\bullet\lambda_{23}]+[P4(t)\bullet\lambda_{43}]+[P5(t)\bullet\lambda_{53}]]$$

$$dP4(t)/dt=[[(\lambda_{41}+\lambda_{42}+\lambda_{43}+\lambda_{44}+\lambda_{45})P4(t)\}]+[P1(t)\bullet\lambda_{13}]+[P2(t)\bullet\lambda_{24}]+[P3(t)\bullet\lambda_{33}]+[P5(t)\bullet\lambda_{52}]]$$

$$dP5(t)/dt=[[-(\lambda_{51}+\lambda_{52}+\lambda_{53}+\lambda_{54}+\lambda_{55})P5(t)\}]+[P1(t)\bullet\lambda_{14}]+[P2(t)\bullet\lambda_{22}]+[P3(t)\bullet\lambda_{34}]+[P4(t)\bullet\lambda_{42}]]$$

$$dP5_1(t)/dt=[[-(\lambda_{51}+\lambda_{52}+\lambda_{53}+\lambda_{54}+\lambda_{55}+\lambda_{5_16}+\lambda_{5_17})\Delta t\}]+[P1(t)\bullet\lambda_{14}]+[P2(t)\bullet\lambda_{22}]+[P3(t)\bullet\lambda_{34}]$$
$$+[P4(t)\bullet\lambda_{42}]+[P5_2(t)\bullet\lambda_{5_26}]+[P5_3(t)\bullet\lambda_{5_36}]]$$

$$dP5_2(t)/dt=[[-(\lambda_{51}+\lambda_{52}+\lambda_{53}+\lambda_{54}+\lambda_{55}+\lambda_{5_26}+\lambda_{5_27})P5_2(t)]+[P1(t)\bullet\lambda_{14}]$$
$$+[P2(t)\bullet\lambda_{22}]+[P3(t)\bullet\lambda_{34}]+[P4(t)\bullet\lambda_{42}]+[P5_1(t)\bullet\lambda_{5_17}]+[P5_3(t)\bullet\lambda_{5_37}]]$$

$$dP5_3(t)/dt=[[-(\lambda_{51}+\lambda_{52}+\lambda_{53}+\lambda_{54}+\lambda_{55}+\lambda_{5_36}+\lambda_{5_37})P5_3(t)]+[P1(t)\bullet\lambda_{14}]$$
$$+[P2(t)\bullet\lambda_{22}]+[P3(t)\bullet\lambda_{34}]+[P4(t)\bullet\lambda_{42}]+[P5_1(t)\bullet\lambda_{5_16}]+[P5_2(t)\bullet\lambda_{5_27}]]$$

$$dP6(t)/dt=[[P1(t)\bullet\lambda_{15}]+[P2(t)\bullet\lambda_{25}]+[P3(t)\bullet\lambda_{35}]+[P4(t)\bullet\lambda_{45}]+[P5(t)\bullet\lambda_{55}]]$$

© 2013, IJARCSSE All Rights Reserved

In the matrix form

For Figure4 (a)

$$\begin{bmatrix} dP1(t)/dt \\ dP2(t)/dt \\ dP3(t)/dt \\ dP4(t)/dt \\ dP5(t)/dt \\ dP6(t)/dt \end{bmatrix} = \begin{bmatrix} [-(\lambda_{11}+\lambda_{12}+\lambda_{13}+\lambda_{14}+\lambda_{15})] & \lambda_{21} & \lambda_{31} & \lambda_{41} & \lambda_{51} & 0 \\ \lambda_{11} & [-(\lambda_{21}+\lambda_{22}+\lambda_{23}+\lambda_{24}+\lambda_{25})] & \lambda_{32} & \lambda_{44} & \lambda_{54} & 0 \\ \lambda_{12} & \lambda_{23} & [-(\lambda_{31}+\lambda_{32}+\lambda_{33}+\lambda_{34}+\lambda_{35})] & \lambda_{43} & \lambda_{53} & 0 \\ \lambda_{13} & \lambda_{24} & \lambda_{33} & [-(\lambda_{41}+\lambda_{42}+\lambda_{43}+\lambda_{44}+\lambda_{45})] & \lambda_{52} & 0 \\ \lambda_{14} & \lambda_{22} & \lambda_{34} & \lambda_{42} & [-(\lambda_{51}+\lambda_{52}+\lambda_{53}+\lambda_{54}+\lambda_{55})] & 0 \\ \lambda_{15} & \lambda_{25} & \lambda_{35} & \lambda_{45} & \lambda_{55} & 0 \end{bmatrix} * \begin{bmatrix} P1(t) \\ P2(t) \\ P3(t) \\ P4(t) \\ P5(t) \\ P6(t) \end{bmatrix}$$

Matrix for Figure4 (b)

$$\begin{bmatrix} dP5_1(t)/dt \\ dP5_2(t)/dt \\ dP5_3(t)/dt \end{bmatrix} = \begin{bmatrix} \lambda_{14} & \lambda_{22} & \lambda_{34} & \lambda_{42} & -(\lambda_{51}+\lambda_{52}+\lambda_{53}\ \lambda_{54}+\lambda_{54}+\lambda_{5_1 6}+\lambda_{5_1 7})] & \lambda_{5_2 6} & \lambda_{5_3 6} & 0 \\ \lambda_{14} & \lambda_{22} & \lambda_{34} & \lambda_{42} & \lambda_{5_1 7} & -(\lambda_{51}+\lambda_{52}+\lambda_{53}\ \lambda_{54}+\lambda_{54}+\lambda_{5_2 6}+\lambda_{5_2 7})] & \lambda_{5_3 7} & 0 \\ \lambda_{14} & \lambda_{22} & \lambda_{34} & \lambda_{42} & \lambda_{5_1 6} & \lambda_{5_2 7} & -(\lambda_{51}+\lambda_{52}+\lambda_{53}\ \lambda_{54}+\lambda_{54}+\lambda_{5_3 6}+\lambda_{5_3 7})] & 0 \end{bmatrix} * \begin{bmatrix} P1(t) \\ P2(t) \\ P3(t) \\ P4(t) \\ P51(t) \\ P5_2(t) \\ P5_3(t) \\ P6(t) \end{bmatrix}$$

or $\underline{P}$= [A] • $\underline{P}$ where [A] is defined as state transition matrix.

Assuming failure rate & Probability at time t then use them for finding probability at time (t+Δt)

Δt=0.06

For state P1, P2, P3, P4, P5, P6

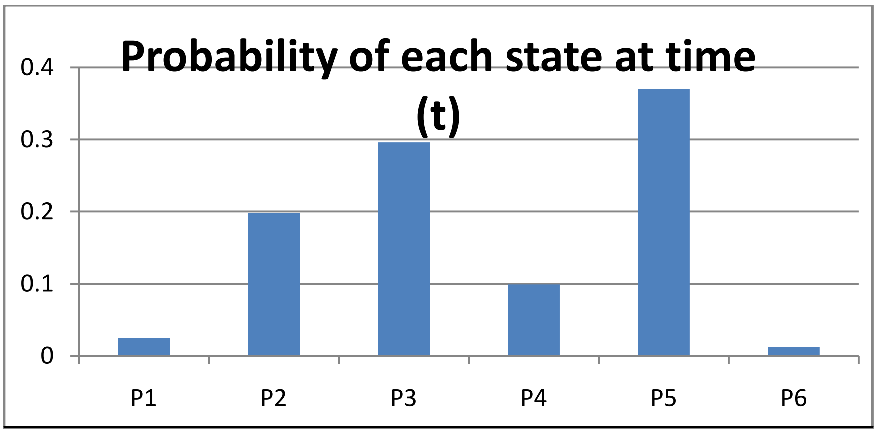
$$\begin{bmatrix} dP1(t)/dt \\ dP2(t)/dt \\ dP3(t)/dt \\ dP4(t)/dt \\ dP5(t)/dt \\ dP6(t)/dt \end{bmatrix} = \begin{bmatrix} -0.975 & 0.075 & 0.042 & 0.05 & 0.067 & 0 \\ 0.198 & -0.95 & 0.333 & 0.1 & 0.333 & 0 \\ 0.296 & 0.375 & -0.967 & 0.5 & 0.2 & 0 \\ 0.099 & 0.125 & 0.217 & -0.95 & 0.133 & 0 \\ 0.370 & 0.25 & 0.292 & 0.25 & -0.8 & 0 \\ 0.012 & 0.125 & 0.083 & 0.05 & 0.067 & 0 \end{bmatrix} * \begin{bmatrix} 0.025 \\ 0.198 \\ 0.296 \\ 0.099 \\ 0.370 \\ 0.012 \end{bmatrix}$$

After Matrix Mathematics:

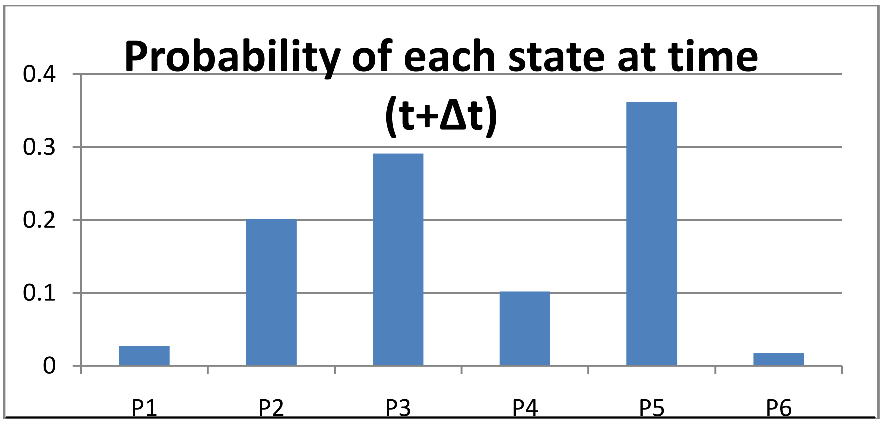
P1(t+Δt)	=	0.027
P2(t+Δt)	=	0.201
P3(t+Δt)	=	0.291
P4(t+Δt)	=	0.102
P5(t+Δt)	=	0.362
P6(t+Δt)	=	0.017

© 2013, IJARCSSE All Rights Reserved

Result:



Graph (a) showing probability of each state at time (t)



Graph (b) Showing Probability of each state at time(t+Δt)

CONCLUSION:

Software is an essential component of many safety critical systems. These systems depend on the reliable operation of software components. Software reliability is the probability of fault free operation of software components in a specified period of time in a specified environment. A number of studies have been conducted for measuring reliability of given software as a result a number of analytical models has been introduced. One of them is the Markov Model that utilizes Markov Property & makes use of software architecture which is defined as the set of components, connectors & configurations i.e. it is an architecture based software reliability model. In this paper we have developed Markov Model of a web application used in the real world. It consists of 6 states and 30 transitions then derive matrix for it . The important thing is that by use the derived matrix the analyst needs only to generate the states & transition between states as defined by the element failure rates. This information can then be inputted into the matrix and compute the state probabilities using matrix mathematics.

REFERENCES

[1] http://en.wikipedia.org/wiki/Markov_chain.

[2] Norman B. Fuqua *"The Applicability of Markov Analysis Methods to Reliability, Maintainability & Safety"* Volume10, Number2

[3] http://en.wikipedia.org/wiki/Markov_model

[4] http://en.wikipedia.org/wiki/Software_Reliability_Testing

[5] Chaitanya Kallepalli, Jeff Tian "*Measuring and Modelling Usage and Reliability for Statistical Web Testing"* IEEE Transactions On Software Engineering, VOL. 27, NO. 11, NOVEMBER 2001.

[6] Tomohiko Takagi, Zengo Furukawa "*Construction Method of a High-Order Markov Chain Usage Model*" Published by 2007 IEEE.

[7] Zhao Li and Jeff Tian "*Testing the Suitability of Markov Chains as Web Usage Models"*

[8] J. D. Musa. *Software Reliability Engineering*. McGraw-Hill, New York, 1998

[9] J.A. Whittaker and M.G. Thomason "A Markov Chain Model for Statistical Software Testing" IEEE Trans. Software Eng., vol. 20, no. 10, pp. 812±824, Oct. 1994.

[10] J.A. Whittaker and J.H. Poore, "Markov Analysis of Software Specifications", *ACM Transactions on Software Engineering and Methodology*, Vol.2, No.1,pp.93-106, 1993.

[11] G.H. Walton, J.H. Poore, and C.J. Trammell, "Statistical Testing of Software Based on a Usage Model", *Software Practice and Experience*, Vol.25, No.1, pp.97-108, 1995.

© 2013, IJARCSSE All Rights Reserved