Implementation Of Architecture Based Object Oriented Software System Reliability Prediction Model : Research Paper

Shailja Kashyap[1], Durgesh Srivastava[2]

[1]Student M.Tech (CSE), BRCM-CET, Bahal, India. 9461059381

[2]Assistant Professor, CSE Department, BRCM-CET, Bahal, India.

09996745993

ABSTRACT

Reliability of software system is the measure of extent to which a system can work efficiently and as expected, subjected to certain constraints and time period .Prediction or measurement of reliability of software system have been in lime light in world of software engineering researches .Multiple models and measurement tools have been designed to estimated the reliability based on different criteria for example, some models are based on failure rate, some on complexity matrix and others on system component failure probabilities .In this paper we introduce a system or software tool to predict object oriented software reliability at design and analysis phase .System is an implementation of already existing architecture based model , which makes use of annotated UML diagrams .The system output helps the user to easily calculate various failure probabilities and at the end over all system reliability ,at design phase itself, so the appropriate steps could be taken at an early stage .UML diagrams are created using UMLET software and the system have been developed in CPP .

KEYWORDS

Software reliability, UML, Measurement tool, Measurement methods, Object oriented systems.

1. INTRODUCTION

IEEE 610.12-1990 defines reliability as "The ability of a system or component to perform its required functions under stated conditions for a specified period of time." Whereas, Quality is the totality of features and characteristics of a product or a service that bears on its ability to satisfy the given needs [2] .From these two definitions from literature it could be derived that quality of a software is directly proportional to its reliability where quality is the dependent variable and reliability is the independent variable .Though quality of a software depends on various other factors ,such as GUI ,performance ,ease-of-use etc. ,reliability is one of the major factors .Quality of software is highly desirable in competitive world of software ,so the reliability prediction and measurement has been in lime light since decade .Many models have been proposed based on different theories and at different phases of SDLC to estimate the reliability of software systems .Most of the models proposed ,estimate reliability at system level ,which demands lot of testing work to be done for the integrated system .In this paper ,we chose to implement a system model that works on system architecture at design and analysis phase to predict the system reliability at a very early stage of SDLC thus saving lots of resources and time as compared to other model.

2. SYSTEM MODEL

Model follows architecture based technique, in which reliability is measured at design and analysis time using annotated UML diagrams for example .Use case, Sequence, Deployment diagrams etc. . Using this model the reliability of functional requirements are measured .In [1][3][4] authors, proposed a model which can estimate the system reliability as early as system architecture is available at design and analysis phase of SDLC. The model implements Bayesian reliability prediction algorithm proposed in paper [1][3].

2.1 Assumptions of model from [1][3][4]

1. In future COTS components would be sold with a specification sheet having their reliability details.
2. If any component fails system will fail.
3. Independence of failure among components.

2.2 Equations derived from model

Probability of executing an use case is given by

$$P(j) = \sum_{i=1}^{m} q_i \cdot p_{ij} \quad j = 1 \text{ to } n \qquad 2.1$$

Where,

q_i, is the probability that user u_i will use the system requesting some functionality.

m and n are the number of users and use cases respectively

p_{ij} is probability of i^{th} user requesting functionality in use case j. There could be number of sequence diagrams for each use case. Taking in account frequency of execution of each sequence diagram, equation (2.2) can be modified as

$$P(k_j) = p(j).f_j(k) \qquad (2.2)$$

Where, $f_j(k)$ is the frequency of execution of k^{th} sequence diagram in j^{th} use case.

Let Θ_i be the failure probability of component then failure probability of component in its respective use case can be calculated using equation

Θ_{ij} = probability(failure of component c_i in scenario j)

$$= 1 - (1- \Theta_i)^{bp_{ij}} \qquad (2.3)$$

bp_{ij} is the busy period of component i in scenario j .

Each pair of components say (l,m) interacting through connector i is subjected to a failure probability say Ψ_i

$$\Psi_{lmj} = (1- \Psi_i)^{(interact\ (lmj))} \qquad (2.4)$$

Where,

$\Psi_{lmj,}$ is the reliability probability of communication between the components.

| (interact (lmj) | , is the number of interactions between the pair of components (l,m) in use case j.

Whole system reliability is given by

$$\Theta_s = 1 - \sum_{i=1}^{k} p_j \left(\prod^{n} (1- \Theta_i)^{bpij} \,.\, \prod (1-\psi_{lij})^{(interact()lij)} \right) \qquad (2.5)$$

3. SYSTEM IMPLEMENTATION

System designed and developed is based on already existing model proposed in [1][3], described briefly in above section. It also takes information from our previous review paper [5] .It is a console application.

3.1 System Requirements : -

- System could be run on any computer system with window 7/8/9/10/vista , Unix platform .
- It requires memory size of 2 MB ,for its installation and execution .
- Annotated UML diagrams are created using UMLET software .

3.2 Example with system execution screenshots

3.2.1 Annotated use case diagram

A use case diagram represents system modules or functionalities, called as use case, and the users of the system called Actors. In this example we have two use cases and two actors, fig 3.1 shows the use case diagram for this particular example.

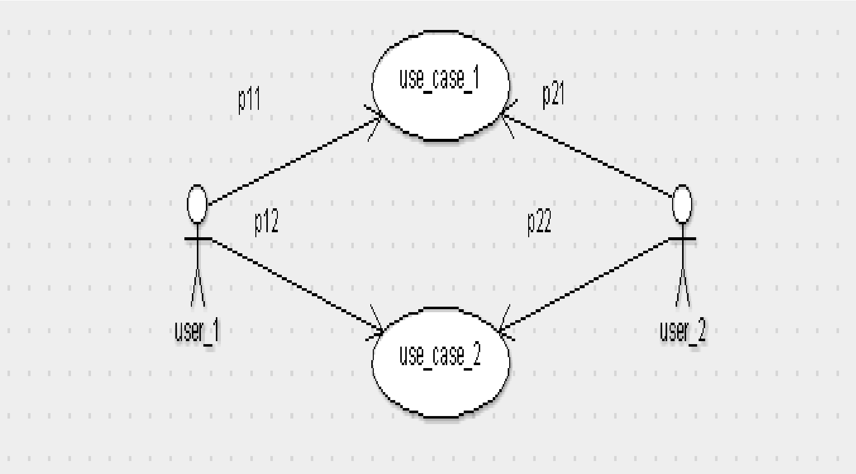
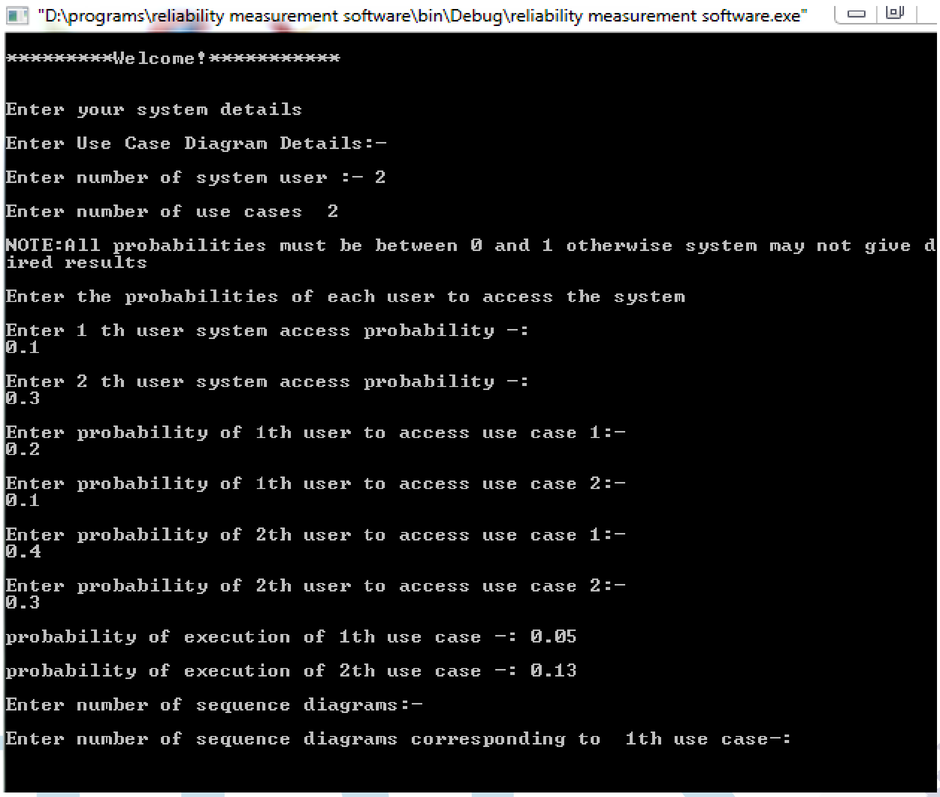


Fig 3.1

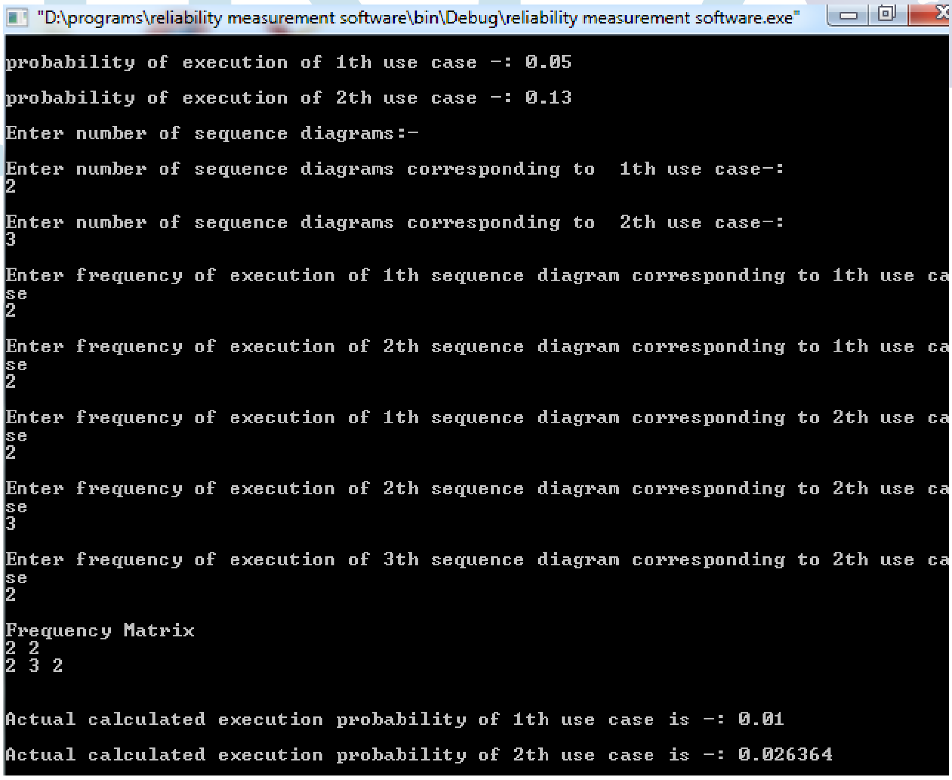
In fig 3.2 and fig 3.3, inputs are submitted from use case diagram, like number of use cases, number of users, system access probabilities of all the users, number of sequence diagram in each use case, frequency of execution of each sequence diagram in their respective use case, and the probability of executing an use case is calculated by the system based on the model discussed in previous section.

3.3.1.2 User inputs and outputs Screenshots.



```
**********Welcome!************

Enter your system details

Enter Use Case Diagram Details:-

Enter number of system user :- 2

Enter number of use cases  2

NOTE:All probabilities must be between 0 and 1 otherwise system may not give d
ired results

Enter the probabilities of each user to access the system

Enter 1 th user system access probability -:
0.1

Enter 2 th user system access probability -:
0.3

Enter probability of 1th user to access use case 1:-
0.2

Enter probability of 1th user to access use case 2:-
0.1

Enter probability of 2th user to access use case 1:-
0.4

Enter probability of 2th user to access use case 2:-
0.3

probability of execution of 1th use case -: 0.05

probability of execution of 2th use case -: 0.13

Enter number of sequence diagrams:-

Enter number of sequence diagrams corresponding to  1th use case-:
```

Fig 3.2



```
probability of execution of 1th use case -: 0.05

probability of execution of 2th use case -: 0.13

Enter number of sequence diagrams:-

Enter number of sequence diagrams corresponding to  1th use case-:
2

Enter number of sequence diagrams corresponding to  2th use case-:
3

Enter frequency of execution of 1th sequence diagram corresponding to 1th use ca
se
2

Enter frequency of execution of 2th sequence diagram corresponding to 1th use ca
se
2

Enter frequency of execution of 1th sequence diagram corresponding to 2th use ca
se
2

Enter frequency of execution of 2th sequence diagram corresponding to 2th use ca
se
3

Enter frequency of execution of 3th sequence diagram corresponding to 2th use ca
se
2

Frequency Matrix
2 2
2 3 2


Actual calculated execution probability of 1th use case is -: 0.01

Actual calculated execution probability of 2th use case is -: 0.026364
```

Fig 3.3

3.2.2 Annotated Sequence Diagram

Sequence diagrams depict how groups of components interact to accomplish a given task. Sequence diagrams provide the sequence in which the events will occur, and can provide the specific information about the timing required for reacting to events. In fig 3.4, fig 3.5 and fig 3.6, user inputs sequence diagram details like number of components, busy periods of each component, and the failure probabilities of each components and the system calculates the failure probability of each component in their respective use case.

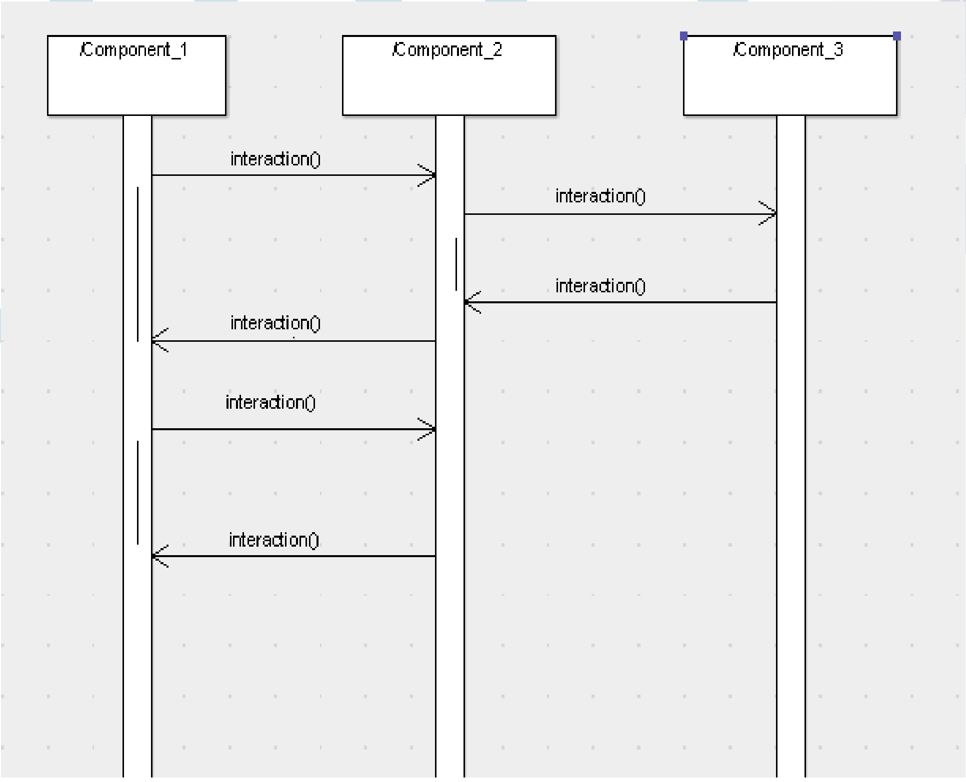
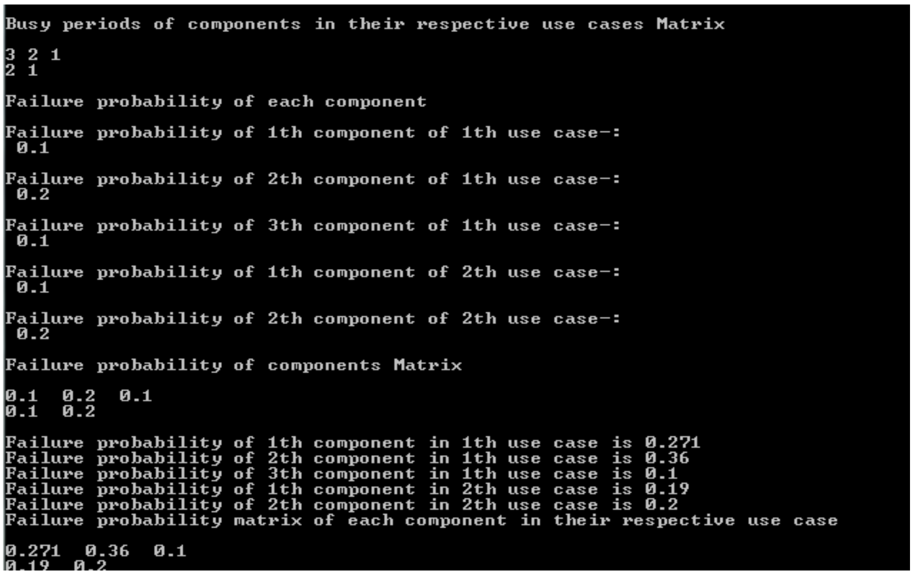


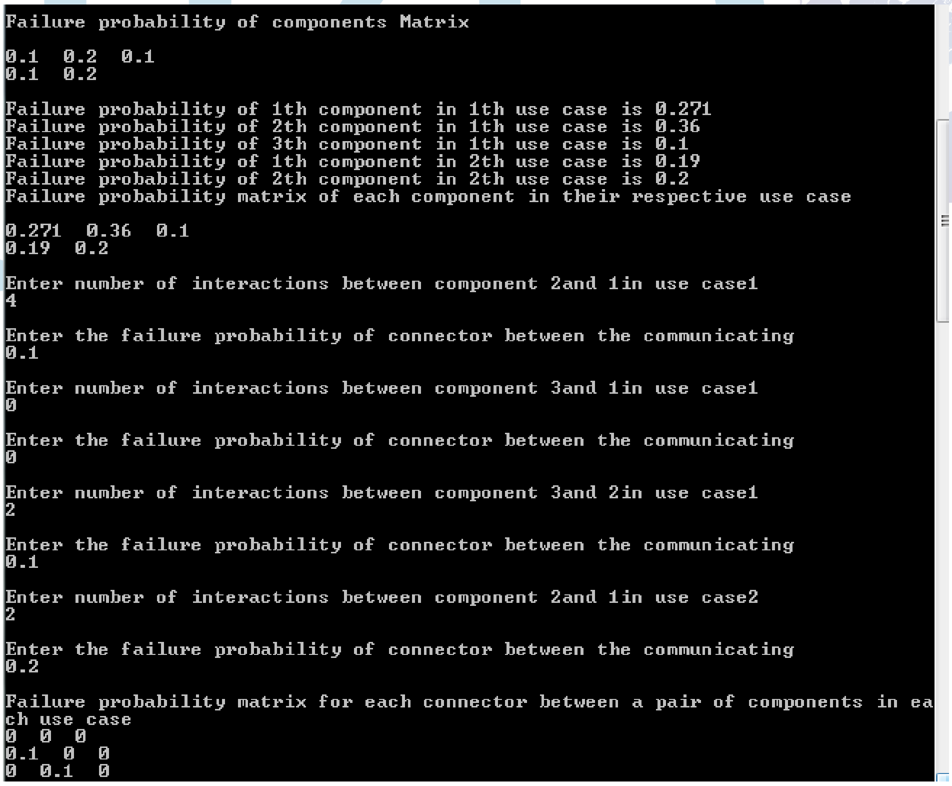
Fig 3.4

3.2.2.1 User inputs and outputs screenshot



```
Busy periods of components in their respective use cases Matrix

3 2 1
2 1

Failure probability of each component

Failure probability of 1th component of 1th use case-:
 0.1

Failure probability of 2th component of 1th use case-:
 0.2

Failure probability of 3th component of 1th use case-:
 0.1

Failure probability of 1th component of 2th use case-:
 0.1

Failure probability of 2th component of 2th use case-:
 0.2

Failure probability of components Matrix

0.1  0.2  0.1
0.1  0.2

Failure probability of 1th component in 1th use case is 0.271
Failure probability of 2th component in 1th use case is 0.36
Failure probability of 3th component in 1th use case is 0.1
Failure probability of 1th component in 2th use case is 0.19
Failure probability of 2th component in 2th use case is 0.2
Failure probability matrix of each component in their respective use case

0.271  0.36  0.1
0.19  0.2
```

Fig 3.5



```
Failure probability of components Matrix

0.1  0.2  0.1
0.1  0.2

Failure probability of 1th component in 1th use case is 0.271
Failure probability of 2th component in 1th use case is 0.36
Failure probability of 3th component in 1th use case is 0.1
Failure probability of 1th component in 2th use case is 0.19
Failure probability of 2th component in 2th use case is 0.2
Failure probability matrix of each component in their respective use case

0.271  0.36  0.1
0.19  0.2

Enter number of interactions between component 2and 1in use case1
4

Enter the failure probability of connector between the communicating
0.1

Enter number of interactions between component 3and 1in use case1
0

Enter the failure probability of connector between the communicating
0

Enter number of interactions between component 3and 2in use case1
2

Enter the failure probability of connector between the communicating
0.1

Enter number of interactions between component 2and 1in use case2
2

Enter the failure probability of connector between the communicating
0.2

Failure probability matrix for each connector between a pair of components in ea
ch use case
0  0  0
0.1  0  0
0  0.1  0
```

Fig 3.6

3.3.3 System Deployment diagram

Deployment diagrams show the platform configuration where the software application is targeted to run. Where the nodes represent platform sites and additional boxes represent software components and are placed into the respective sites they are supposed to be loaded [1][3].

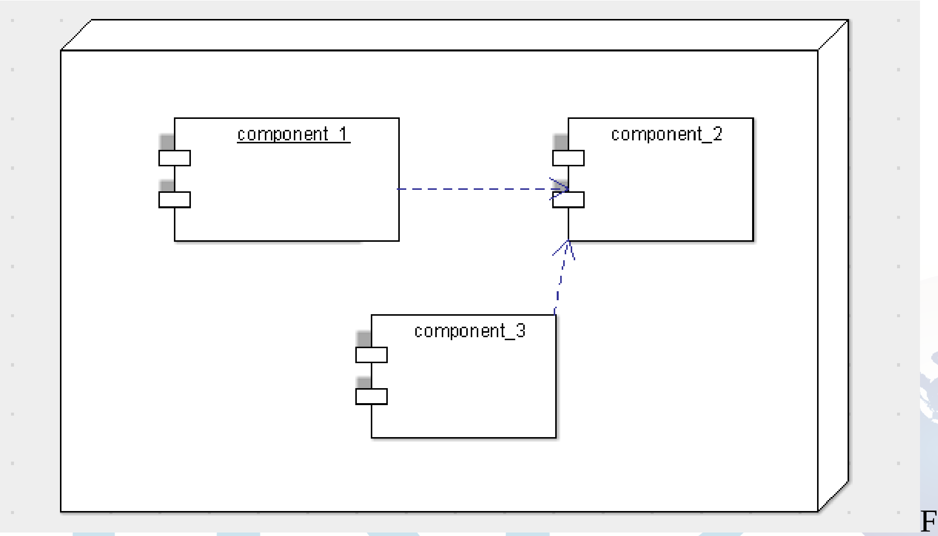
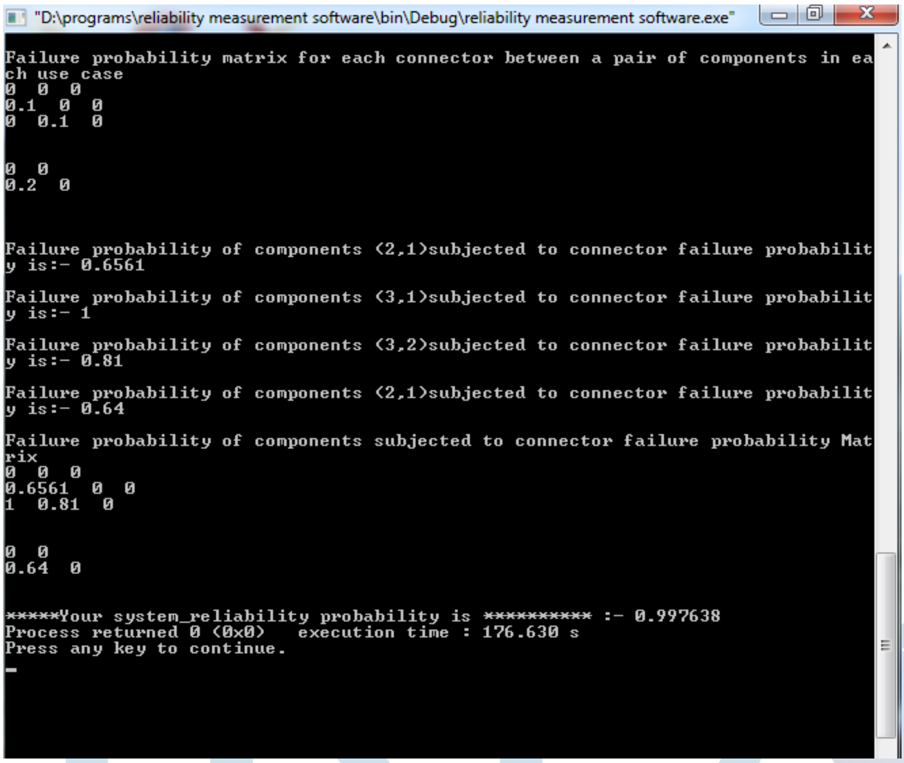


Fig 3.6

In fig 3.6 and 3.7, user inputs the number of interactions between each pair of components and the failure probabilities of connector between them. The system calculates the failure probability of each pair of components subjected to failure probability of connector in particular scenario. Further the system calculates the overall system reliability from above calculated probabilities based on system model.

3.3.3.1 User inputs and outputs Screenshot



```
Failure probability matrix for each connector between a pair of components in ea
ch use case
0  0  0
0.1  0  0
0  0.1  0


0  0
0.2  0



Failure probability of components (2,1)subjected to connector failure probabilit
y is:- 0.6561

Failure probability of components (3,1)subjected to connector failure probabilit
y is:- 1

Failure probability of components (3,2)subjected to connector failure probabilit
y is:- 0.81

Failure probability of components (2,1)subjected to connector failure probabilit
y is:- 0.64

Failure probability of components subjected to connector failure probability Mat
rix
0  0  0
0.6561  0  0
1  0.81  0


0  0
0.64  0


*****Your system_reliability probability is ********** :- 0.997638
Process returned 0 (0x0)   execution time : 176.630 s
Press any key to continue.
_
```

Fig 3.7

4. CONCLUSION

This paper implements the Bayesian reliability prediction model .Software system have been designed and developed which helps user to predict any system reliability based on its UML diagrams. It is easy to use and works efficiently .Multiple number of experiments have been conducted to prove its efficiency and correctness.

5. FUTURE WORK

A lot of future work can be associated be this research work. As it could be seen that system lacks graphical user interface, so just to make it look more pleasant, front end can be designed in any visual programming language. Inputs can be automatically extracted from annotated UML diagrams by the system.

6. REFERENCES

[1] Vittorio Cortellessa, Harshinder Singh, Bojan Cukic. West Virginia University (USA). " A bayesian approach to reliability prediction and assessment of component based systems" . IEEEXplore , 2001.

[2] Ronan Fitzpatrick "Software quality: Definitions and strategic issues " 1996.

[3] Dromey, R.G.: A Model for Software Product Quality. IEEE Transaction on Software Engineering 21(2), 146–162 (1995).

[4] Zhao Hui-Qun, Sun Jing, Gao Yuan. "A new model for estimating the reliability of software system based on components",Northeastern University Qin-Huang-Dao Campus 066004 P.R.China.

[5] Shailja Kashyap, Durgesh Srivavstava."Object oriented software system reliability measurement techniques: A review". International Journal for research and development Organization Sept 2016.

[6] Vittorio Cortellessa, Harshinder Singh, Bojan Cukic. West Virginia University (USA). Early reliability assessment of UML based software models. WOSP. ISBN 1-1-58113-563-7 02/07. July 24-26, 2002.

[7] K.K. Aggarwal, Yogesh Singh. Software Engineering, Third Edition. New Age International Publications. ISBN(10), ISBN(13). 2008.