Online ISSN : 0975-4172
Print ISSN : 0975-4350

Global Journal

of Computer Science and Technology: C

Software & Data Engineering

Data Link Layer Designing

Web-Based Hierarchical Workgroup

Highlights

Trends in Software Testing

Object Oriented Programming

Discovering Thoughts, Inventing Future

VOLUME 14 ISSUE 8 VERSION 1.0

Global Journal of Computer Science and Technology: C
Software & Data Engineering

Global Journal of Computer Science and Technology: C
Software & Data Engineering

Volume 14 Issue 8 (Ver. 1.0)

Open Association of Research Society

The opinions and statements made in this book are those of the authors concerned. Ultraculture has not verified and neither confirms nor denies any of the foregoing and no warranty or fitness is implied.

Engage with the contents herein at your own risk.

The use of this journal, and the terms and conditions for our providing information, is governed by our Disclaimer, Terms and Conditions and Privacy Policy given on our website http://globaljournals.us/terms-and-condition/menu-id-1463/

By referring / using / reading / any type of association / referencing this journal, this signifies and you acknowledge that you have read them and that you accept and will be bound by the terms thereof.

All information, journals, this journal, activities undertaken, materials, services and our website, terms and conditions, privacy policy, and this journal is subject to change anytime without any prior notice.

Incorporation No.: 0423089
License No.: 42125/022010/1186
Registration No.: 430374
Import-Export Code: 1109007027
Employer Identification Number (EIN):
USA Tax ID: 98-0673427

Global Journals Inc.

*(A Delaware USA Incorporation with "Good Standing"; **Reg. Number: 0423089**)*

Sponsors: Open Association of Research Society
Open Scientific Standards

Publisher's Headquarters office

Global Journals Headquarters
301st Edgewater Place Suite, 100 Edgewater Dr.-Pl,
Wakefield MASSACHUSETTS, Pin: 01880,
United States of America
USA Toll Free: +001-888-839-7392
USA Toll Free Fax: +001-888-839-7392

Offset Typesetting

Global Journals Incorporated
2nd, Lansdowne, Lansdowne Rd., Croydon-Surrey,
Pin: CR9 2ER, United Kingdom

Packaging & Continental Dispatching

Global Journals
E-3130 Sudama Nagar, Near Gopur Square,
Indore, M.P., Pin:452009, India

Find a correspondence nodal officer near you

To find nodal officer of your country, please email us at *local@globaljournals.org*

eContacts

Press Inquiries: *press@globaljournals.org*
Investor Inquiries: *investors@globaljournals.org*
Technical Support: *technology@globaljournals.org*
Media & Releases: *media@globaljournals.org*

Pricing (Including by Air Parcel Charges):

For Authors:
22 USD (B/W) & 50 USD (Color)
Yearly Subscription (Personal & Institutional):
200 USD (B/W) & 250 USD (Color)

Integrated Editorial Board
(Computer Science, Engineering, Medical, Management, Natural Science, Social Science)

President Editor (HON.)

Chief Author (HON.)

Dean & Editor-in-Chief (HON.)

Contents of the Issue

Global Journal of Computer Science and Technology: C
Software & Data Engineering
Volume 14 Issue 8 Version 1.0 Year 2014
Type: Double Blind Peer Reviewed International Research Journal
Publisher: Global Journals Inc. (USA)
Online ISSN: 0975-4172 & Print ISSN: 0975-4350

Data Link Layer Designing Issues: Error Control–A Roadmap

By Monika Singh & Ruhi Saxena

Mody University, India

Abstract- Different networks are used to transfer data from one device to another with acceptable accuracy. For most applications, a system must guarantee that the data received are identical to the data transmitted. Transmission media are most error-prone link. In a network, the capacity of nodes is different and the rate at which the sender is sending data might not be the same rate at which the receiver accepts it. In this paper, we discuss on designing issues of data link layer. The primary focus ison various error detecting and controlling mechanisms.

Keywords: error detection, LRC, VRC, CRC, checksum.

GJCST-C Classification : C.2.5 C.2.6

Data Link Layer Designing Issues: Error Control—A Roadmap

Monika Singh [α] & Ruhi Saxena [σ]

Abstract- Different networks are used to transfer data from one device to another with acceptable accuracy. For most applications, a system must guarantee that the data received are identical to the data transmitted. Transmission media are most error-prone link. In a network, the capacity of nodes is different and the rate at which the sender is sending data might not be the same rate at which the receiver accepts it. In this paper, we discuss on designing issues of data link layer. The primary focus ison various error detecting and controlling mechanisms.

Keywords: error detection, LRC, VRC, CRC, checksum.

I. Introduction

Data Link Layer is the second layer of OSI Layered Model. Data link layer is responsible for converting data stream to signals bit by bit and to send that over the underlying hardware. At the receiving end, Data link layer picks up data from hardware which are in the form of electrical signals assembles them in a recognizable frame format, and hands over to upper layer [2][6].

Data link layer has two sub-layers:

- Logical Link Control : Deals with protocols, flow-control and error control.
- Media Access Control : Deals with actual control of media.

Data link layer does many tasks on behalf of upper layer. The main functionalities are:

a) Framing

Data-link layer takes packets from Network Layer and encapsulates them into Frames. Then, sends each Frame bit-by-bit on the hardware. At receiver's end Data link layer picks up signals from hardware and assembles them into frames.

b) Addressing

Data-link layer provides layer-2 hardware addressing mechanism. Hardware address is assumed to be unique on the link. It is encoded into hardware at the time of manufacturing.

c) Synchronization

When data frames are sent on the link, both machines must be synchronized in order to transfer to take place.

d) Error Control

Sometimes signals may have encountered problem in transition and bits are flipped. These errors are detected and attempted to recover actual data bits. It also provides error reporting mechanism to the sender.

e) Flow Control

Stations on same link may have different speed or capacity. Data-link layer ensures flow control that enables both machines to exchange data on same speed.

f) Multi-Access

Hosts on shared link when tries to transfer data, has great probability of collision. Data-link layer provides mechanism like CSMA/CD to equip capability of accessing a shared media among multiple Systems.

II. Designing Issues In Data Link Layer

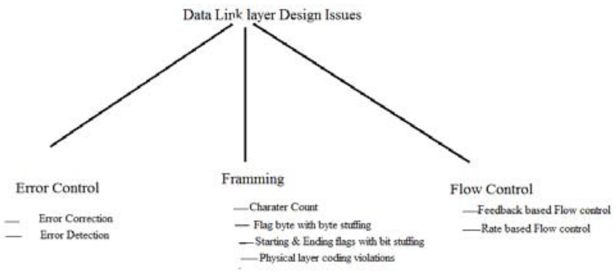

Data Link layer Design Issues
Error Control
Error Correction
Error Detection
Framming
Charater Count
Flag byte with byte stuffing
Starting & Ending flags with bit stuffing
Physical layer coding violations
Flow Control
Feedback based Flow control
Rate based Flow control

Figure 1 : Designing issues in data link layer

The designing issues of data link layer are following:

a) Error Control

- Error control includes both error detection and error correction.
- It allows the receiver to inform the sender if a frame is lost or damaged during transmission and coordinates the retransmission of those frames by the sender.
- Error control in the data link layer is based on automatic repeat request (ARQ). Whenever an error is detected, specified frames are retransmitted [4][7].

b) Framing

- Break down a stream of bits into smaller, digestible chunks called frames
- Allows the physical media to be shared

Author α σ : FET, Mody University of Science and Technology, India.
e-mails: Dhariwal.monika@gmail.com, ruhi.saxena2011@gmail.com

- Multiple senders and/or receivers can time multiplex the link
- Each frame can be separately addressed

- Provides manageable unit for error handling
    - Easy to determine whether something went wrong
    - And perhaps even to fix it if desire

c) *Flow Control*

- Flow control coordinates the amount of data that can be sent before receiving acknowledgement
- It is one of the most important functions of data link layer.
- Flow control is a set of procedures that tells the sender how much data it can transmit before it must wait for an acknowledgement from the receiver.
- Receiver has a limited speed at which it can process incoming data and a limited amount of memory in which to store incoming data.
- Receiver must inform the sender before the limits are reached and request that the transmitter to send fewer frames or stop temporarily.
- Since the rate of processing is often slower than the rate of transmission, receiver has a block of memory (buffer) for storing incoming data until they are processed

III. Error Control

Error controls are the techniques that enable reliable delivery of digital data over unreliable communication channels. Many communication channels are subject to channel noise, and thus errors may be introduced during transmission from the source to a receiver.

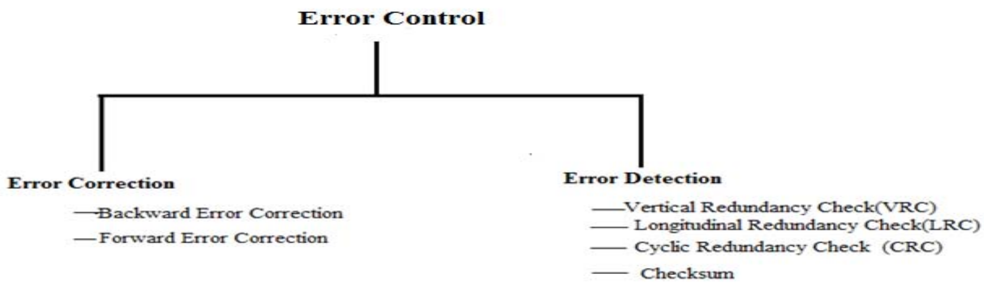

Error Control
Error Correction
Backward Error Correction
Forward Error Correction
Error Detection
Vertical Redundancy Check(VRC)
Longitudinal Redundancy Check(LRC)
Cyclic Redundancy Check (CRC)
Checksum

Figure 2 : Error Control Methodologies

Generally, there are two types of Error:

Single bit error: Single-bit error means that only one bit of data been change through transportation of data [7]. It changes either from 0 to 1 or 1 to 0. This one bit changed cannot be ignored since one bit change can change the whole meaning of the data that is transmitted. Figure-3 shows an example of this type of error.

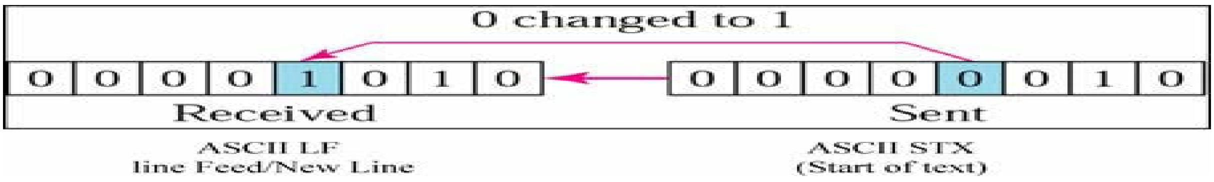


Figure 3 : Single-bit error

Burst error: Burst error means that two or more bits are changed when the transmitting data from the sender to the receiver the data units have change from 0 to 1 or 1 to 0 because of the channel interference [8]. Figure-4 shows the burst error of 8-bits.Burst errors are likely to occur rather that the single-bit error. The duration of the error was longer that duration of 1 bit, which means the data is affected by the noise usually affect a set of bits. The number of bits that corrupted always depends on the data rate and duration of noise.

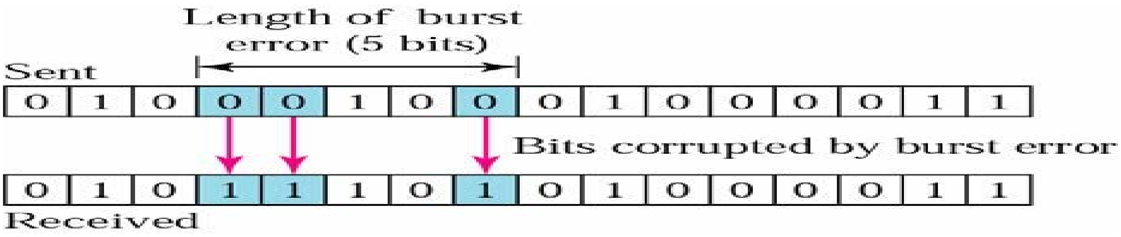


Figure 4 : Burst error of length 8

a) *Error Correction*

In digital world, error correction can be done in two ways:

- *Backward Error Correction:* When the receiver detects an error in the data received, it requests back the sender to retransmit the data unit.
- *Forward Error Correction:* When the receiver detects some error in the data received, it uses an error-correcting code, which helps it to auto-recover and corrects some kinds of errors.

The first one, Backward Error Correction, is simple and can only be efficiently used where retransmitting is not expensive, for example fiber optics. But in case of wireless transmission retransmitting may cost too much. In the latter case, Forward Error Correction is used [5].

To correct the error in data frame, the receiver must know which bit (location of the bit in the frame) is corrupted. To locate the bit in error, redundant bits are used as parity bits for error detection. If for example, we take ASCII words (7 bits data), then there could be 8 kind of information we need. Up to seven information to tell us which bit is in error and one more to tell that there is no error.

b) *Error Detection*

Error detection means to decide whether the received data is correct or not without having a copy of the original message.Error detection uses the concept of redundancy, which means adding extra bits for detecting errors at the destination.

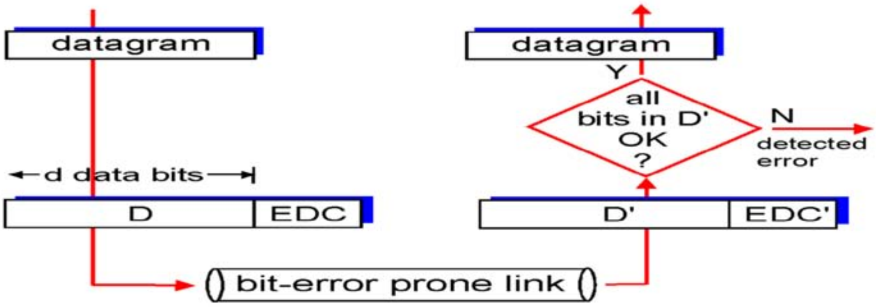


Figure 5 : Error Detection Mechanism

Where D = Data, EDC = Error Detection Code (redundancy)

There are four types of redundancy checks:

- VRC (Vertical Redundancy Check).
- LRC (Longitudinal Redundancy Check).
- CRC (Cyclical Redundancy Check).
- Checksum

i. *Vertical Redundancy Check (VRC)*

It is also known as parity check. There are two types of parity check schemes: even and odd parityChecks [8]. In an even parity check scheme, the sender simply includes one additional bit and choose its value such that the total number of 1's in the d+1 bits (the original information plus a parity bit) is even. For odd parity check scheme, the parity bit value is chosen such that there is odd number of 1's. Table-1 shows a table contains 3 bits string. The transmitter will add 0 or 1 to the bits string according to the parity check mechanism (even or odd). When the receiver receives the bits string, the receiver will use the same mechanism to count the 1's in the bit string to determine whether it matches the counted parity from the transmitter or not [1].

Table 1 : Parity bits that are compute for bits string

3 bits string	Odd parity	Even parity
000	1	0
001	0	1
010	0	1
100	0	1
110	1	0
111	1	0

This scheme can detect only single bits. So if two or more bits are changed then that cannot be detected.

ii. *Longitudinal Redundancy Checksum (LRC)*

Longitudinal Redundancy Checksum is an error detecting scheme which overcomes the problem of two erroneous bits. In this concept of parity bit is used but with slightly more intelligence. With each byte we send one parity bit then send one additional byte which has the parity corresponding to the each bit position of the sent bytes. So the parity bit is set in both horizontal and vertical direction. If one bit gets flipped we can tell which row and column have error then we find the intersection of the two and determine the erroneous bit. If 2 bits are in error and they are in the different column and row then they can be detected. If the errors are in the same column then the row will differentiate and vice versa. Parity can detect the only odd number of errors. If they are even and distributed in a fashion that in all direction then LRC may not be able to find the error [2][5].

iii. *Cyclic Redundancy Checksum (CRC)*

In this method, a sequence of redundant bits, called the CRC or the CRC remainder, is appended to the end of the data unit so that the resulting data unit becomes exactly divisible by a second, predetermined binary number. At its destination, the incoming data unit is divided by the same number. If at this step there is no remainder, the data unit assumes to be correct and is accepted, otherwise it indicates that data unit has been damaged in transmission and therefore must be rejected. The redundancy bits are used by CRC are derived by dividing the data unit by a predetermined divisor. The remainder is the CRC.

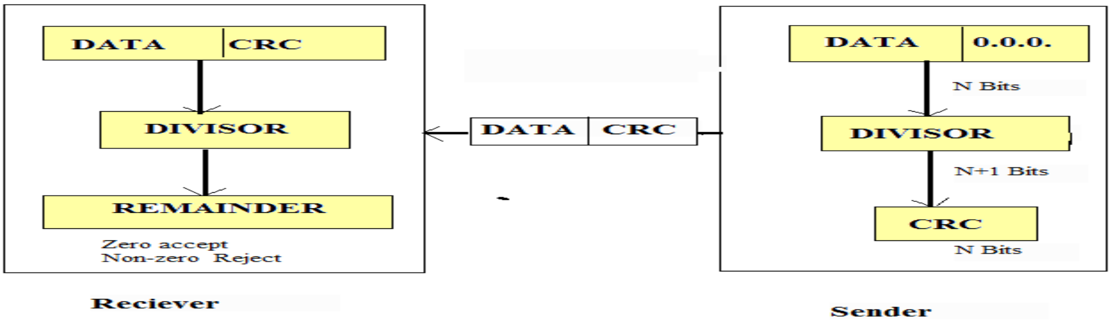


Figure 6 : Mechanism of CRC at sender and Receiver side

For example, the CRC generator at the sender side:

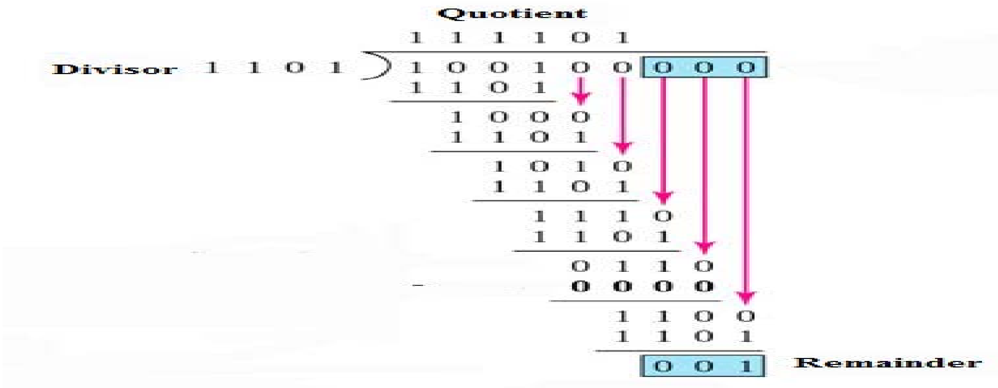


Figure 7 : CRC in sender side

The CRC checker at receiver side:

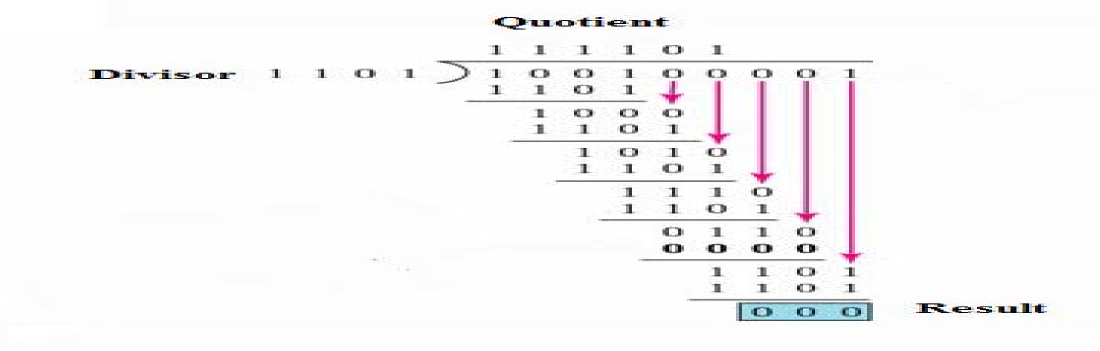


Figure 8 : CRC in receiver side

iv. *Checksum*

The checksum is based on the concept of redundancy. As shown in Figure-9, in the sender, the checksum generator subdivides the data unit into equal segments of n bits (usually 16). These segments are added using ones complement arithmetic in such a way that the total is also n bits long. That total (sum) is then complemented and appended to the end of the original data unit as redundancy bits, called the checksum field. The extended data unit is transmitted across the network. So if the sum of the data segments is T, the checksum will be -T. The receiver performs the same calculation on the received data and compares the result with the received checksum. If the result is 0, the receiver keeps the transmitted data; otherwise, the receiver knows that an error occurred discards the transmitted data [9].

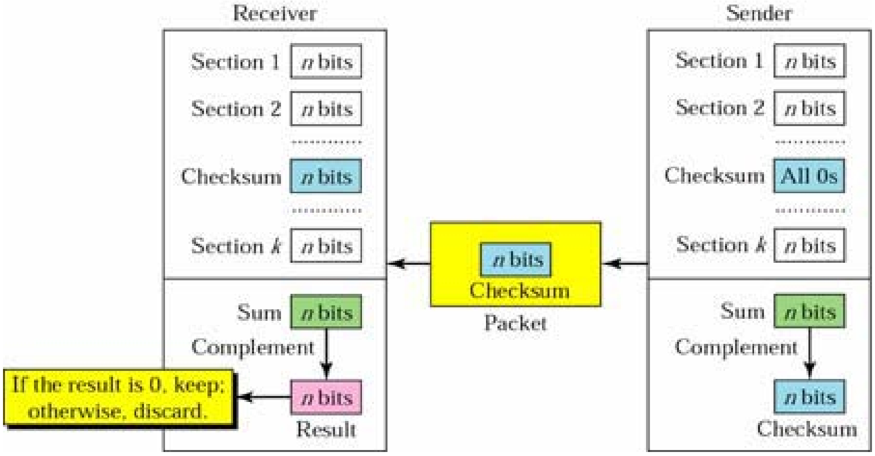


Figure 9 : Working of Checksum at sender and Receiver side

For example, the following block of 16 bits is to be sent using a checksum of 8 bits

10101001 00111001

The numbers are added using one's complement
10101001
00111001
Sum 11100010
Checksum00011101

The pattern sent is

10101001 00111001 00011101

At the receiver end, same calculation has been done on the received data and compares the result with the received checksum in the following way:
10101001
00111001
00011101
Sum 11111111
Checksum 00000000

When the receiver adds the three sections, it will get all 1s, which, after complementing, is all 0s and shows that there is no error.

IV. Conclusion

There are various methods of detecting error in the data link layer. Every method of error detectioncan detect error accurately and effectively. Every method has its own advantage and its own mechanism to detect error. VRC is simple and can detect all single-bit error. CRC has a very good performance in detecting single-bit errors, double errors, an odd number of errors, and burst errors while checksum is not asefficient as the CRC.

References Références Referencias

1. Dimitri Bertsekas and Robert Gallager, "Data Networks".
2. Larry Peterson and Bruce Davie, "Computer Networks: A Systems Approach".
3. http://www.cse.iitk.ac.in/users/dheeraj/cs425/lec09.html/
4. http://ecomputernotes.com/computernetworkingnotes/communication-networks/what-is-error-correction-and-detection.
5. Forouzan, "Data Communications and Networking".
6. J. Chellis, C. Perkins and M. Strebe, "MCSE: Networking Essentials Study Guide with CDROM" , 2nd Ed. 1999.
7. F. Behrouz and M. Firouz, "Computer Networks: A Top Down Approach", 1st International Ed. McGraw-Hill., 2012.
8. J. Fletcher, "An arithmetic checksum for serial transmissions, Communications". IEEE Transactions on. 30: 247-252, 1982.
9. L. L. Peterson and B. S. Davie, "Computer networks: a systems approach", 2007, 3rd Ed. Elsevier.

This page is intentionally left blank

Global Journal of Computer Science and Technology: C
Software & Data Engineering
Volume 14 Issue 8 Version 1.0 Year 2014
Type: Double Blind Peer Reviewed International Research Journal
Publisher: Global Journals Inc. (USA)
Online ISSN: 0975-4172 & Print ISSN: 0975-4350

An Overview of Recent Trends in Software Testing

By Anupama Surendran & Philip Samuel

Cochin University of Science & Technology India

Abstract- In the field of search based software testing, genetic algorithm based testing has received a major share of attention among researchers during the last few years. Though there are advantages for this type of testing, there also exist some practical difficulties which can make this technique less attractive for software testing industry. The potential of program slicing in testing has not been fully exploited till now and the works that have explicitly demonstrated the application of slicing in testing field are rare. Our paper aims to analyze existing techniques for software testing and to introduce an approach for software testing using program slicing technique. A systematic review of genetic algorithm based works reveals that, fitness function design, population initialization and parameter settings impact the quality of solution obtained in software testing using genetic algorithm. Based on the conclusions from the existing literature, we have probed deeper about the issues in these areas. Making an unbiased review like this may help to solve these unresolved issues in genetic algorithm based software testing. In this work, we have emphasized and has given clear directions on how slicing can be used as a potential tool for practical software testing. In addition, a set of research questions have been framed, which may be answered by reviewing the study made in this work. This may help future research in this area, leading to major breakthrough in software testing field.

Keywords: program slicing, software testing, forward slicing, genetic algorithms.

GJCST-C Classification: D.2.5

An Overview of Recent Trends in Software Testing

Anupama Surendran [α] & Philip Samuel [σ]

Abstract- In the field of search based software testing, genetic algorithm based testing has received a major share of attention among researchers during the last few years. Though there are advantages for this type of testing, there also exist some practical difficulties which can make this technique less attractive for software testing industry. The potential of program slicing in testing has not been fully exploited till now and the works that have explicitly demonstrated the application of slicing in testing field are rare. Our paper aims to analyze existing techniques for software testing and to introduce an approach for software testing using program slicing technique. A systematic review of genetic algorithm based works reveals that, fitness function design, population initialization and parameter settings impact the quality of solution obtained in software testing using genetic algorithm. Based on the conclusions from the existing literature, we have probed deeper about the issues in these areas. Making an unbiased review like this may help to solve these unresolved issues in genetic algorithm based software testing. In this work, we have emphasized and has given clear directions on how slicing can be used as a potential tool for practical software testing. In addition, a set of research questions have been framed, which may be answered by reviewing the study made in this work. This may help future research in this area, leading to major breakthrough in software testing field.

Keywords: program slicing, software testing, forward slicing, genetic algorithms.

I. Introduction

In God we trust, everything else we test . This famous quote conveys the idea that almost all the things in this world are unreliable without testing [6]. Proper testing makes the software robust and trustworthy and hence the importance of testing cannot be overemphasized. From simple home appliances and common automobiles, to life support devices like mechanical ventilators and mission critical systems like nuclear reactors, there is an unending list of components which depend on some form of software for their proper functioning [27]. These softwares in turn depend on testing for their infallibility. Imagine a pharmaceutical company introducing a new drug in the market without proper trials and testing. It is not only illegal, but also extremely unsafe and potentially deleterious. Similarly, software development without testing makes it unreliable, unusable and even unsafe.

While one of the main intentions of software testing is to check for and identify errors in software, a software tester has a much wider gamut of responsibilities. For example in our real life activity, in an automobile where there is a sound due to the loosening of wheel, the defect may be corrected by tightening it, but the alignment of the tightened wheel may not be synchronous with the other wheels. Therefore in the next step, the wheels are to be aligned for the proper running of the vehicle. Similarly, finding the root cause or in other words, finding the dependency during software testing is one of most challenging aspects of software testing as rectifying an error may introduce some side effects in the software. Getting the dependency relations present in a program serves as the backbone of several other processes in software development, such as regression testing, program comprehension, maintenance, reverse engineering and re-engineering [16, 17]. This implies that, though software testing can be very challenging, it has a very significant influence and marked relevance in software development industry. In the earlier days, most of the applications used simple software and they were mostly standalone applications. The nature of modern day software can make its testing not an easy task. Many of the software used nowadays is real-time and embedded software with web interface. This type of software may have several interconnected modules and such software needs to be continuously tested until they get outdated from the world market. Technological changes, requirement changes and platform changes raise the need for continuing testing in such systems. In such software, the software dependency consideration is an unavoidable factor which decides the reliability of the software. Even a minor error may cause great mishap in such software applications. The unrestricted size of the source code is another problem plaguing the software testing industry. In the case of large commercial software, there will be several modules and lines of code which make software testing process more difficult. As testing cost increases with source code size, it should be one of the primary concerns of the software tester. In the field of software testing, a software tester cannot leave the scene after finishing the testing process [31]. During software testing, the test cases designed for solving the error in some part of the source code may prove to be insufficient to solve the bugs occurring some other parts of the source code. This is similar to the creation of mutant species. For example, long term use of an insecticide against a particular species of insect, makes it vulnerable to development of

Author α σ : Cochin University of Science & Technology, Kochi, Kerala, India. e-mail: anudeepaknair@gmail.com

resistance by genetic modification and mutation in the insect. In such a situation, new insecticides have to be used to kill that insect. Similarly, the test cases designed for a particular test scenario may fail in some situation. This may be due to the changes made in the source code or due to the change in design requirements made as per user specifications. New test cases are to be found for solving such problem or the existing test cases should be updated by the software tester. From the above discussion it is evident that, a good tester should be a good software designer, an intuitive code developer and a reliable maintenance person, all rolled into one. For example, consider the situation where a company decides to change its product as per user requirements. Now, the software designer and code developer can fulfill their parts just by completing the work in their respective areas of expertise. On the other hand, for the testing to be fully reliable, the tester has to understand the changes made by designer and code developer and then develop appropriate testing methods. Truly speaking, a good software tester has to be a skilled all-rounder.

Several methods were developed with an aim to address the challenges existing in software testing industry. Among the different software testing strategies, search based testing has received immense attention and especially, genetic algorithm based testing has made a marked influence in software testing research [30]. This is due to the adaptability of genetic algorithms to handle the testing process and the ability to represent the software testing problem as an optimization problem [38]. Considering the volume of work done in genetic algorithm based software testing, it is crucial to identify the merits and demerits of this approach. Even though genetic algorithm based testing has made a great impact in academic research, only very little attention has been given to understand the complexities of using genetic algorithms in practical software testing. This work focuses on this and we have tried to highlight the challenges involved in genetic algorithm based approaches for using it as a practical tool in software testing. The main reason for choosing this problem in our work is because of the usage of genetic algorithms in software testing without knowing the ambiguities in genetic algorithm based testing. In this paper, we have mentioned some works which utilize genetic algorithm for testing [38, 39, 40, 44, 51, 52, 54]. We can see that none of these works have adopted any general operator setting for testing purpose. The inherent non-deterministic nature of the genetic operators makes the program testing process a demanding task. The strength of using genetic algorithm mainly depends on setting the genetic parameters to their appropriate values and this in turn depends on the problem to be solved. This itself is a major challenge faced by testers.

In this work, we have suggested a program slicing approach for software testing and have highlighted the strengths of using program slicing as a tool in software testing industry. It was Weiser who introduced slicing in 1979 [15, 53] and his work encouraged many research works developing slicing algorithms. According to Weiser, slicing criterion consists of two parameters and it is represented as (V, n), where 'V' is a set of variables and 'n' is the program point [53]. In program slicing, source code size is minimized by converging focus on some specific program part implied by the 'slicing criterion' [20,49]. This property of slicing is highly relevant, as source code size is a major concern is modern day software. Instead of analyzing the whole program, slicing reduces the program search space which in turn minimizes the testing effort. Setting the slicing criterion with respect to the variable with incorrect value can help to identify the portion of source code which causes error during program testing. Here the manual effort of the program tester is reduced considerably as there is no need to consider the whole source program [11, 47]. Slicing also helps to trace program dependencies which are very crucial in testing. In several works it has been mentioned that program slicing may be used for testing purpose [17, 20, 21]. None of these works gave a clear picture of how to utilize slicing to make testing more meaningful. Apart from program testing, slicing can be used in several applications such as program debugging [34, 53], program comprehension [22] and program maintenance [17]. In this paper, we have demonstrated a forward slicing approach for testing and have tried to mark the merits of program slicing based testing approaches.

Finally, this paper aims to:

- Introduce program slicing as a major research direction in software testing
- Present an analytical description of program slicing and to demonstrate how it can be applied in software testing
- Assess the current research trends in software testing with a special focus on genetic algorithm based testing
- Analyze the shortcomings and challenges for making genetic algorithm based approaches practical in software testing industry
- Highlight the significance of program dependency in software testing, and explain how program slicing can effectively resolve this issue

The remaining section of the paper is organized as follows. Section 2 gives the basics of program slicing and genetic algorithm. Section 3 compares program slicing based testing and genetic algorithm based testing approaches. Based on the observations made in section 3, some research questions are framed in section 4. In section 5, we have given an explanation of the research questions in section 4. Threats to validity of this work are given in section 6 and section 7 gives the conclusion.

II. Basics

As we are doing a detailed study of genetic algorithm based and program slicing based software testing methods, we shall go through the basic principles of genetic algorithm and program slicing concepts. Based on the conclusions from the exiting literature, we will have to probe deeper about the issues in these areas. Making an unbiased review like this may help to solve the issues in genetic algorithm based software testing and at the same time help to understand the relevance of program slicing in software testing. This may help the future researchers working in this area.

a) Genetic Algorithms

In order to conduct a proper review of genetic algorithm based software testing, it is essential that one should be familiar with the basic concepts and terms in genetic algorithm. This is dealt with in this section. Genetic algorithm is a type of evolutionary algorithm and is considered as the best and the strongest of all evolutionary algorithms [18, 24]. It is a type of search technique developed by John Holland and works on Darwin‗s principle of survival of the fittest. Genetic algorithm uses the technique of natural genetics, representing a computer model of biological evolution. Genetic algorithms have the ability to solve a variety of optimization and search problems. Several testing techniques use genetic algorithms believing that testing may be carried out in a better way using the natural evolutionary process present in them [39].

Genetic algorithm identifies an optimal solution for a problem by applying natural evolutionary techniques to a group of possible solutions referred to as —population [18, 40]. After each generation, a new generation is formed which is better than the previous generation. The series of steps involved in genetic algorithm are population initialization, selection, crossover, mutation and termination. A string of digits called chromosomes are present and each individual of the string is called a gene. Each individual in the population has a fitness value which decides the quality and performance of that individual. Greater the fitness value better will be the problem solving capacity of an individual [25]. Collection of chromosomes makes up a population. The initial population is created randomly and the fitness of the individuals in the population is calculated. This information is used to select the best candidates for forming the next generation parents. After selecting parents of the successive generation, the next step is to combine these candidates to form the offspring. Crossover operation is used to perform this [36, 54]. Crossover enables the selection of good features from parents to form the offspring. Mutation is applied to the offspring to create better quality individuals. Mutation is defined as the process of altering the genes in the chromosome [43]. A new generation is chosen from the offspring based on the fitness of the individuals. These individuals are considered as parents of the next generation. This cycle is repeated until a global solution for the problem is obtained. The basic steps of genetic algorithm are given in algorithm 1.

ALGORITHM 1

```
procedure Genetic Algorithm
begin
GET (Initial Population);
CALCULATE FITNESS (Initial Population)
loop
FINALZE POPULATION FOR CROSSOVER (Parent population)
PERFORM CROSSOVER (Parent population, child)
APPLY MUTATION (Child)
CALCULATE FITNESS (Child)
GET NEXT GENERATION (Parent population, Child)
stop process when TERMINATION CRITERA
exit loop
end
```

b) Program Slicing

This section deals with some of the common terms in program slicing. Slicing is defined as the process of deleting all those statements from a program which cannot affect the values of a variable of interest. In other words, a slice is a subset of source program statements. Slicing is performed based on slicing criteria. A slicing criterion comprises a program location and a set of variables known as slice set. If P is a program, x is a statement in P and y is a variable in P, then the slicing criterion (C) is given as C= (x, y). Program slicing can be divided into various types. Based on slicing criteria, the two main types are static and dynamic slicing [32, 35], while based on direction of slicing the two main types are forward and backward slicing [22, 49].

i. Static Slicing

A slice constructed by ignoring those parts of the program that are not relevant to the values stored in the chosen set of variables at the chosen point is known as static slice [8, 34]. As mentioned above slicing criterion C= (x, y), where x is a statement in the P (program) and y is a variable in P. Given a variable ‘y‘ and a point of interest ‘x‘, slice will be constructed for y at x. An example program is given in table 1, where the static slice criterion is given as <11, a>. The result will be the set of statements <4, 5, 6, 8, 9>. Backward slicing gives all the program statements which affect the value of a particular variable at a particular point [TIP 1995]. Forward slicing gives all the program statements which are affected by declaring a variable at a given point in the program [22, 29].

Table 1 : Static slicing

Program Statements	Static slice for criterion <11, a>
1 main()	4 cin>> b;
2 {	5 a = 0;
3 int a,b;	6 while (b <= 10)
4 cin>> b;	8 a=a+b;
5 a = 0;	9 ++ b;
6 while (b <= 10)	
7 {	
8 a=a+b;	
9 ++ b;	
10}	
11 cout<< a;	
12 cout<< b;	
13 }	

ii. *Dynamic Slicing*

The concept of dynamic slicing was given by Korel [33]. The set of statements that affect the value of a variable for one specific input is known as dynamic slice. In dynamic slicing we have to consider three parameters. First one is the point of interest within the program, second one is the variable and the third one is the sequence of input values for which the program was executed. Dynamic slicing criterion is defined as C= (x, y, i). Here x is the statement in the program, y is the subset of variables in the program and i is the input value [11]. A sample program to be sliced is given below in table 2. The variable with respect to which slicing is to be done is p, slicing point is the end of the program and input given is n=0.

Table 2 : Dynamic slicing

Program Statements	Dynamic Slicing Criterion :-(10, p, n=0,)
1 scanf("%d",&n);	p=0
2 s=0;	
3 p=0;	
4 while (n>0)	
5 {	
6 s=s+n;	
7 p=p*n;	
8 n=n–1;	
9 }	
10 printf ("%d%d", p, s);	

In static slicing though the size of the slices obtained will be large, all possible executions will be considered. On the other hand, in dynamic slicing the down side of small size of slices is that the result will be focused only for a specific input [32].

III. Evaluation of Testing Approaches

This section analyses the testing approach based on genetic algorithm and introduces our approach based on program slicing. Here we have identified some points to justify our analysis and these are used to frame the research questions in section 4. We have divided this section into three parts. In the first part the purpose of software testing is explained. The second part deals with genetic algorithm based software testing. Some relevant works in that field and our observations regarding genetic algorithm based testing are given in this section. In the third part we have introduced our program slicing based testing approach and have described its benefits and importance.

a) *Software testing*

The section gives an insight into the basics of software testing. In software testing the target program is executed to identify the errors. This is followed by debugging to rectify the identified errors [21]. Before starting the testing process, the objectives or the goals should be properly set and the tester should be aware of the strategy to be followed to achieve the set goals [10]. It is very essential that the tester should have an idea of user requirements and should also be able to identify the conditions which will have an adverse effect on the selected testing strategy. The main objectives of testing are [4, 41]

- To affirm that the software developed is error free
- To check whether the developed software is functioning correctly according to the program developer and program tester
- To confirm that the developed software works correctly without causing any data loss.

Therefore developing an effective method for testing is an inevitable part of all software systems

b) Genetic algorithm based testing

In the past few years, search based software testing, especially evolutionary algorithm, has gained immense popularity [2, 9]. A graph is shown in figure 1, which shows an increase in rate of publications and research works in search based software testing during the period 1975 to 2010[37]. Among evolutionary algorithms, genetic algorithm is one of the widely researched techniques for software testing. They are included in dynamic testing techniques [26]. In dynamic testing, the program is executed based on given input data to obtain the corresponding output, while in static testing, the program has to be analyzed line by line to check for the errors in the program. Thus in static testing, the ability to find errors depends on the tester's experience.

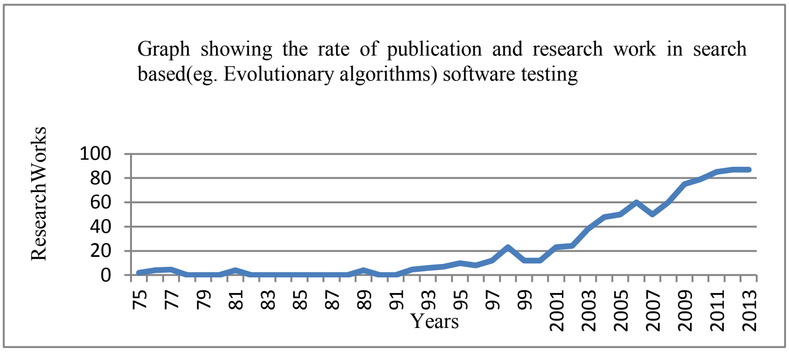


Figure 1 : Research works in search based software testing during the period 1975 to 2013

Genetic algorithms are used to perform automated software testing due to their ability to represent the testing problem as an optimization function. Finding a solution for this optimization problem gives a solution for the testing process also. There were several attempts to generate test data using single population, multiobjective, master-slave, fine-grained and coarse-grained genetic algorithms [1, 9]. We have limited our literature review to some of the most relevant works which have used the concepts of genetic algorithm and single objective fitness function in testing. A detailed study of these works is done to make an assessment of genetic algorithm based software testing approach. In the next paragraph, we discuss some of the most relevant works in genetic algorithm based software testing.

A path wise test data generation using genetic algorithms was introduced by Pei et al. [45]. A control flow graph was constructed and the paths were manually selected from the graph. Only two loops were covered at a time. They designed the fitness function based on the paths selected from the graph. Genetic algorithm based testing was used by Roper et al. [46] for testing C program. They used the branch coverage criteria. In their approach, a random method for population selection was used and this population was subjected to crossover and mutation to generate better individuals. A branch coverage criterion was used by Jones et al. [30] in their work for generating test data using genetic algorithms. Hamming distance approach was used to design the fitness function and their approach could cover programs which contain up to three loops. Pargas et al. [44] developed a tool called TGen which uses genetic algorithm for program testing. A parallel processing approach was used in TGen to improve the testing process. A path coverage and branch coverage approach was used in TGen. The performance of TGen was compared with a tool called Random which is a tool based on random method. Test cases which covered the largest number of predicates were given the highest fitness values. Bueno et al. [7] developed a method for software testing using genetic algorithms. They used the path coverage criteria and introduced the path similarity metric as fitness function. The population initialization was made by checking the previous nature of the population. This helps to create better individuals in the successive generations. Wegener et al. [52] used a statement coverage criterion during testing and they introduced a fitness function which is decided based on the approximation level and normalized predicate level distance. Michael et al. [2001] developed a tool called GADGET which uses genetic algorithms for generating test data for C programs. They designed the fitness function based on some predicate function. Their tool had many limitations like the inability to handle Boolean variables. Doungsa-ard et al. [12] used a genetic algorithm based approach to generate test data for UML state diagrams. They used the transition coverage approach and the fitness function was designed based on the number of transitions fired by the input sequence. The population initialization was made based on the nature of the previous generation individuals. Hermadi et al. [1] used a path coverage criterion to genetic test cases using genetic algorithm. The overall fitness function was a measure of aggregation of individual's fitness function. Table 5 gives a list of some of the works which is uses genetic algorithms for software testing.

A review of these works, throws up some of the pertinent issues in genetic algorithm based software testing. These factors, which play a major role in genetic algorithm based testing and influence its outcome to a significant degree, are given below:

– Population generation
– Design of fitness function
– Response time prediction
– Setting of parameters

i. *Population generation*

This includes initialization and representation of the population, strategies for population selection and the determination of population size. The population which is initialized may itself be the set of initial potential solution. The representation of population is another issue. Population can be represented as a group of 0's and 1's, as a group of integers, as decimal numbers or as characters. In some problems a tree representation is also possible. Based on the problem, appropriate method of representation is applied. Improper representation of the individual in genetic algorithms may cause unexpected variations in the final result [24, 25].

Table 5 : Summary of GA based works on software testing

WORK	COVERAGE	FITNESS FUNCTION	GA TYPE & POPULATION REPRESENTATION	POPULATION SIZE & SELECTION STRATEGY	CROSS OVER TYPE	MUTATION TYPE
DOUNGSA-ARD et al. [2002]	Transition	Number of transitions fired by input sequence	Simple GA & Sequence of triggers	10 & Previous knowledge	Two point	Random mutation & 0.5
HERMADI et al. [2001]	Path	Fitness= Number of violations +Distance	Simple GA &	30 & Roulette wheel selection	Single point	0.1 0r 0.3
WEGENER et al. [2001]	Statement	Approximation level and normalized predicate level distance	Simple & multi population GA & Integer representation	Stochastic universal sampling	Single point	Discrete recombination, 1 & multiple strategies
BUENO et al. [2002]	Path	FT=NC-EP/MEP	Simple GA& Binary string	80 and Selection based on Previous knowledge	Single point	Simple & 0.03
MICHAEL et al. [2001]	Branch	Predicate function	Simple GA & Binary String	24, 100 and Random selection	Single point	Simple & 0.001
PRAGAS et al. [1999]	Statement & Branch	Common predicates	Simple GA & Input data list	100 & Random selection	Single point	Simple & 0.10
JONES et al. [1996]	Branch(Maximum 3 loops)	Hamming distance	Simple GA & Binary plus sign & gray code	45 & Random selection	Uniform	Reciprocal &Weighted. Reciprocal & five least
ROPER et al. [1995]	Branch	Coverage percentage	Simple GA & Character string	User input & Random selection	Single point	Simple mutation. Mutation rate decide by user

PIE et al. [1999]	Path(Maximum 2 loops)	F= C-[10*n+5*n(n-1)/2	Simple GA & Binary string	Program's s size & Random selection	Single point	Simple mutation & 0.001

The next major concerns related to population are the population selection strategy and population size. Either a random method or a heuristic based method is used to initialize the population. In the random method, population is selected randomly. In the heuristic based approach, instead of setting the population randomly, some tests are performed and the individuals are selected based on the test results. This shows that, population selection strategy can be based on several methods to select the appropriate population. The population size can also be a confounding factor because if the population size is too small the genetic algorithm will not search all the possible solution areas to procure an optimal solution [9, 12]. In this case, the individuals may reproduce abundantly and the resulting diversity in population may cause the individuals to converge to a point which appears to be better than the neighboring points. In such a situation, even though there is a chance that a better solution exists, it is missed as the population size is already declared to be very small. This is known as the premature convergence problem [40]. Hence declaring the correct population size still remains a problem in genetic algorithm and research is still ongoing in this area. Before using genetic algorithm for software testing, these inherent issues have to be addressed. Due to the shortcomings of single population genetic algorithm, parallel genetic algorithm has been tried in many applications [30]. Parallel genetic algorithms are similar to single population genetic algorithms running in different machines. The performance of parallel genetic algorithms is affected by the way in which the computers are networked. In effect, even though parallel genetic algorithms may speed up the computation process compared to single population genetic algorithm, several issues in the network implementation topology needs to be dealt with.

We have used the Genetic algorithm solver tool in Matlab 7.8 to give an idea of the population initialization issues presented above. The initial parameter settings for the Genetic algorithm tool are given below.

Genetic algorithm parameter settings
Minimize Objective Function (f) = 10-x
Population type: Double vector
Fitness Scaling: Rank
Crossover type: Scattered
Migration rate: 0.2
Stopping Criteria: 100 generations
Time limit: 10 second

All the parameters except the population size are kept constant. The result obtained for various population sizes is given in Table 6. The objective function value and the value of the best individual present in all iterations are also displayed. From Table 6, it can be inferred that as the population size increases, the result obtained becomes better. Another illustration is given below in figures 2 to 6. These show that, as the population size increases beyond a certain size, the time taken for fitness function optimization increases.

Table 6 : Function values and final point values

Populatio n Size	Best Individual final point value	Objective function value
20	24.76	-14.76
20	24.88	-14.85
30	27.30	-17.30
30	28.37	-18.37
70	44.51	-34.51
70	39.95	-29.95
1000	67.99	-57.99

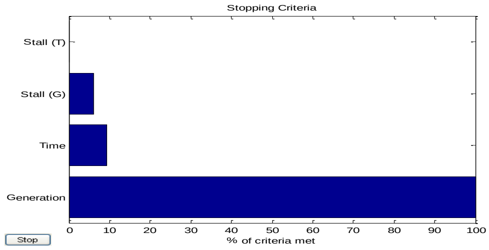


Figure 2 : Stopping criteria for population size =20

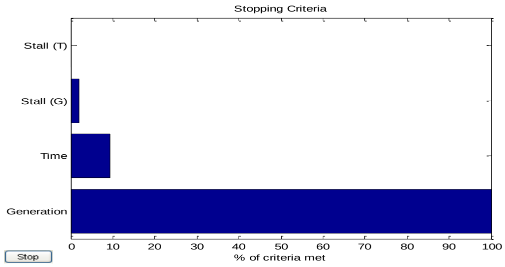


Figure 3 : Stopping criteria for population size =30

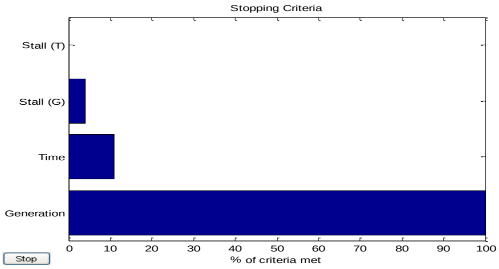


Figure 4 : Stopping criteria for population size =70

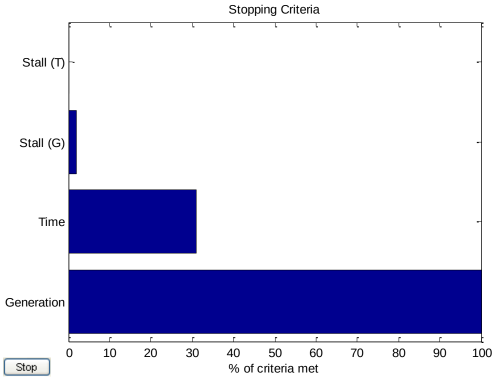


Figure 5 : Stopping criteria for population size =1000

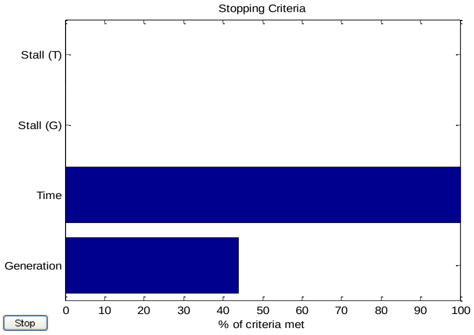


Figure 6 : Stopping criteria for population size =10000

In figures 2 to 6, the time taken to compute the fitness function optimization process for population sizes 20, 30, 70, 1000 and 10000 is showed. Some of the terms related to the fields in these figures are explained below.

- Stopping criteria: Decides the cause of algorithm termination
- Generations: Gives the maximum number of iteration the genetic algorithm runs before termination
- Time limit: Gives the maximum time limit in seconds the genetic algorithm should function before termination
- Stall generations: Genetic algorithm terminates when the weighted average change in the fitness function value over stall generations is less than function tolerance
- Stall time limit: Genetic algorithm terminates when there is no improvement in the fitness value which is the best in a specified time interval
- Function tolerance: The algorithm stops if the weighted average relative change in the best fitness function value over Stall generations is less than or equal to Function tolerance.

When the population size is defined as 20, 30 and 70 respectively, the corresponding fitness values are obtained and the genetic algorithm terminates when the maximum number of generations are exceeded. The time taken for these three processes is almost the same. These can be inferred from the results given in figure 2, 3 and 4. In figure 5 when the population size is 1000, the time taken for fitness function optimization is greater compared to the time taken for population size 20, 30 and 70 and here also the genetic algorithm terminates when the maximum number of generations exceeded the limit specified. In figure 6, it can be seen that only 44 iterations were able to run within the time limit specified as the time limit exceeded the maximum value. Here, an increase population size caused an overrun in time limit. These results point out that population initialization can influence the final result and the population initialization process is problem dependent. For small non- critical optimization problems, the size of the population may not be a critical factor. In critical problems, the population size is very crucial [50].

ii. *Setting of parameters*

In genetic algorithm based program testing, the parameter setting needs special attention. For example in the case of crossover and mutation, their rates should be not be set at either high or low levels. According to the problem‘s nature the parameter settings should be adjusted. The following section gives a description of some of the operator settings used in genetic algorithm based testing.

a. *Selection*

In selection, individuals are selected from the parent population for crossover and mutation to produce next generation individuals [28, 45, 51]. There are different types of selections like roulette wheel, tournament selection, random selection, best selection etc. In roulette wheel selection individuals are selected according to their fitness. Each individual will be assigned a fitness value and the normalized fitness value is calculated. After calculating the normalized fitness value, accumulated fitness value is calculated by adding the fitness value of the concerned individual and the sum of the fitness value of all other individuals. A random number is selected between 0 and 1 and the selected individual will have an accumulated fitness value greater than all other previous individuals but less than the remaining individuals. Tournament selection is a refinement of roulette wheel selection. Here roulette wheel selection is repeatedly applied to produce a group of population and the best individual is selected from this group. In random selection method, the chromosome is selected randomly from the given population whereas in best selection method the individual with the highest fitness value is selected. There are many other types of selection methods, but we have mentioned only a few. There is no specific rule which implies the usage of a particular type of selection method during software testing process. This is one of the greatest difficulties in genetic algorithm based software testing, as the final outcome of testing differs according to the type of selection method used.

b. Crossover

Crossover is the process of combination of parent chromosomes to produce offspring [HOLLAND 1979]. The process of crossover affects the process of test data generation using single population genetic algorithm. The most commonly used types of crossover are one point crossover, two point crossover and uniform crossover. For example consider two parent individuals where the chromosomes are represented as bit strings:

Parent 1:1010101010
Parent 2:1000110000

If the crossover occurs after the sixth bit in the parents, then two children will be formed and the last four bits of both the parents are interchanged. The result can be represented as follows:

Child 1:1010100000
Child 2:1000111010

In uniform crossover, the crossover points are not selected. The parent bits are swapped randomly with 50% probability. If the third, sixth, seventh and tenth bit positions of the parent individuals are swapped, then two children will be produced and they can be represented as follows:

Child 1:1000110010
Child 2:1010101000

By using uniform crossover the diversity in the individuals produced is more compared to single and two point crossover and a better result is obtained. A better result for a given problem may be obtained, even if the testing process is done with the most suitable type of crossover. Solving this uncertainty in genetic algorithm crossover selection still remains as a challenge.

c. Mutation

Mutation is the process of altering the value of genes present in the chromosome for creating genetic diversity [18]. Diversity in the population will create better individuals compared to a population without genetic diversity. According to the problem to be solved, mutation rates can be set to specific values. If the rate of mutation is set to high value, the search will become similar to a random search and if the mutation rate is very low then there will be no diversity in the population. Therefore generally the value of mutation is set between 0.01 and 0.05 [40]. From table 5, we can notice that the mutation rate is set to different values in the listed works. The main problem faced here is that, varying the mutation rate results in a change in the final result and this issue still remains unresolved in genetic algorithm based testing process.

d. Uncertainty in Parameter Settings

Even after testing a program using the best available genetic parameters, a better solution or the same solution can be obtained even if we use less competing methods of crossover, selection and mutation for solving the same problem. This shows the uncertain nature of genetic algorithms [38]. We have some examples to illustrate the uncertainty of genetic algorithms. Our aim is to use genetic algorithms to minimize the SchafferF6 function, which is a published benchmark function. SchafferF6 function is a complex optimization problem whose solution can be obtained by applying genetic algorithm based optimization methods. We have considered SchafferF6 function in our optimization test because this function is a multidimensional function. It is having non-linear and oscillatory nature around the optimal solution [18]. This means that SchafferF6 function is having more than single local optima where the genetic algorithm may get halted.

The SchafferF6 function is defined as:

$$f(x) = 0.5 + \frac{(\sin^2\sqrt{x^2+y^2}) - 0.5}{(1.0+0.001(x^2+y^2))^2}$$

Here function minimization is done using two-point crossover and uniform crossover. Initially the objective function or the fitness function minimization is done using two-point crossover. Then the experiment is repeated again using the same parameter settings. The resultant values are noted in each case. Then the objective function minimization is done using uniform crossover. Here also the experiment is repeated using uniform crossover and the values are noted. The results are shown in the table 7, table 8 and table 9.

It has been said that when uniform crossover is used for solving a problem, not only the result will be better compared to two point crossover, but also the convergence happens faster [30]. From the illustrations given below in tables 7, 8, 9, we can see that this is not true in all the cases. In the first trial, the value of the

SchafferF6 function obtained using uniform crossover is better than two point crossover. Further in this case, the time taken is more compared to two-point crossover. In the second trial, the time taken for minimizing the objective function using uniform crossover is less compared to two-point crossover. Here we can see that the fitness function has lower value when two-point crossover is used. Even though there is a little bit difference in time taken to minimize the function, the quality of the result is better in two- point crossover. In the third trial also the value of the fitness function is better when two-point crossover is used. Here the time take is more when two point crossover is used. When uniform crossover is used in the third trial it can be noticed that the value of the fitness function is greater than the value of the fitness function got when two point crossover is used and this indicates that the quality of test data got using two point crossover is better than the quality of test data obtained using uniform crossover. We can see that the time taken for two-point crossover is more compared to time taken when uniform crossover is used. Even though the convergence takes place faster in uniform crossover, it is not mandatory to get minimal value of the fitness function in all the trials.

Table 7 : Trial 1 values

Parameters	Two-Point Crossover	Uniform crossover
Number of Generations	1070	3184
Time taken in seconds	7.657	26.649
Score	0.001982	0.001758
Fitness function Value	0.265497	0.198465

Table 8 : Trial 2 values

Parameters	Two-Point Crossover	Uniform crossover
Number of Generations	949	749
Time taken in seconds	7.336	6.258
Score	0.003094	0.000808
Fitness function Value	0.257263	0.362636

Table 9 : Trial 1 values

Parameters	Two-Point Crossover	Uniform crossover
Number of Generations	499	145
Time taken in seconds	4.543	1.010
Score	0.001609	0.001124
Fitness function Value	0.167003	0.332225

Form these observations we can conclude that, even though there are some general assumptions about the best methods of crossover, selection and mutation, which are to be used for solving a problem, it may not be possible to decide the best combination of these genetic factors as parameter setting in all the cases [26]. Therefore while using genetic algorithms for program testing; we can make only a few assumptions about the problem which is to be solved. All these make the use of genetic algorithm for effective program testing highly complex and impractical.

iii. *Design of fitness function*

Applying genetic algorithm in program testing requires optimizing the specified fitness function. A fitness function should be designed in such a way that it gives optimal solution for a given problem. Defining the fitness function imprecisely may lead to a wrong solution or may cause the problem to be stuck in the local optima [18, 40]. The misleading nature of fitness function creates several problems. For example, the individuals with lower fitness values may be finalized as the optimal solution even when better individuals exist. This mainly occurs when the population size is smaller, because with a small sized population, the result may get converged at a faster rate than normal. Thus, in a limited population, if one of the individuals surpasses the neighbouring individuals, then that point or individual will be considered as the best solution even when better solutions exist. Considering these local points as the candidate solutions and assigning higher fitness values to them will result in a diversion from the original solution. This results from the inherent weakness of genetic algorithms [40]. A group of researchers used an evolutionary algorithm along with a reprogrammable hardware array and the fitness function was designed to output an oscillating signal. At the final stage of the experiment, the researchers found that the circuit had become a radio receiver which was able to pick up and relay an oscillating signal from the nearby electronic device. Here, there was a deviation from the main goal itself and this was due to the fault in the design of the fitness function [19]. Each one of the many works which use genetic algorithm for software testing has designed their own fitness function [37]. Referring the works given in table 5, we can see that none of the works have used similar type of fitness function. For example, Bueno et al. [7] have used a path similarity metric as fitness function and Michael et al. [40] have used the fitness function based on some predicate function. Even though there are some good methods for fitness function calculation, none of them is universally accepted as the gold standard. The fitness function is designed based on the analysis of a problem [24]. In other words, fitness function is problem dependent and this is one of the hurdles to be surmounted while using a genetic approach in software testing.

iv. *Response time prediction*

Fitness function optimization is a heuristic process and the optimization time and effort varies according to the nature of the problem [2]. Therefore, the exact time required for testing a program cannot be accurately predicted. The time varies as the parameter settings are changed. These can be inferred from the graphical figures 2 to 6. From these figures, it is clear that solving a problem with a lower population size will take less time compared to solving the same problem with a higher population size. Even though this is not a major concern in most of the testing applications, some care has to be taken while using genetic algorithm based testing in safety critical applications. In today's world, the workings of all applications are based on real time software. In real-time system the response time plays a critical role and due to the long computation time and uncertainty in the duration of computation time, genetic algorithms cannot ensure constant response time in all the executions [50]. Therefore before implementing the genetic algorithm based system in the original system, a prototype model checking has to be carried out. As stated above, since the performance of genetic algorithm changes according to the change in the parameter values, using genetic algorithms to solve such real time problems should be done with utmost care.

c) *Software Testing using Program Slicing*

In the previous section we saw an overview of genetic algorithm based software testing. We have also explained some issues which can make genetic algorithm based software testing less practical in testing industry. This section looks into the possibilities of program slicing for software testing.

As mentioned in section 2, the concept of slicing was introduced by Weiser and his works encouraged the application of slicing in several fields like program comprehension [22], testing [20, 21, 47], debugging [33, 34], software maintenance [16], program cohesion [43], refactoring [35], reverse engineering [8] etc. We shall see how it can be used for software testing. In software testing, locating the erroneous statements is the key part. As program slicing deletes all those statements from a program which cannot affect the values of a variable of interest, slicing can make the whole software testing process more manageable. Even though some works have mentioned the use of slicing in testing [3, 5, 20, 21, 47], work that has explicitly shown how program slicing may be applied in software testing is extremely rare to the best of our knowledge. We have mentioned some fundamental works in table 10 which apply slicing for identifying test cases during the various phases of software development life cycle. In these works we can notice that they have either mentioned the need of slicing during regression testing process or during the design phase for identifying test cases before the coding phase. Our work illustrates how test cases may be obtained from slices during the testing phase itself.

Table 10 : Works on Program slicing based software testing

Work	Description
Gupta et. al[1992]	Regression testing using slicing
Binkley[1998]	Incremental regression testing using slicing
Harman et al. [1994]	Mentioned that slicing may be applied during the testing phase by checking whether the program is robust or not
Bates et al.[1993]	Slicing applied to identify statements modified in a program dependence graph during the regression testing phase
Samuel et. al[2009]	Using dynamic slicing to generate test cases form UML activity diagrams

i. *Testing Approach*

We have used a forward slicing approach for program testing. Forward slicing is recommended to locate the parts of the program affected by some modification and the sizes of the forward slices are smaller than that of backward slices in some scenarios [22]. In other words, when testing is done with an aim of identifying the errors caused by wrong input variable declaration, forward slicing is more meaningful than static slicing [22]. If the user is supposed to find errors in the output variable then static slicing is more useful than dynamic slicing. In such scenarios it will be more meaningful to apply forward slicing rather than backward slicing. In forward slicing, if a particular statement is affected by the value of the slice variable which is declared at a particular point, then that statement can be added to the list of slice statements. Otherwise there is no need to update the slice list. The whole process will be continued until slicing is performed for all the required variables. The result of the whole process will be a set of statements. These statements are known as forward slice of a particular variable. The forward slicing algorithm suggested in this work is given in algorithm 2

ALGORITHM 2

```
Input: - Program to be sliced (P)
Slicing Criterion (C)
Output: - Forward Slices (F)
begin
1. while p ≠ Ø, source program not empty
2. get C= (n, V)
// where n is statement number, V is the slicing variable
3. while (n ≠ 0 && n < EOP)
```

```
//where EOP is the end of program
{
4. Store ‘V‘ in ‘L‘
// slicing variable ‘V‘ stored in list ‘L‘
5. if ( VAR (L) € n)
// check whether slice variable ‘V‘ stored in list ‘L‘ is present in statement ‘n‘
{
5.1. if ( n is an element of output statement)
F= F U n
//Store n
// include the statement as a slice
5.2. if (n is an element expression)
{
5.2.1. if (VAR (L) € RHS (EXPR))
{
```

```
F=F U n
// Store n
VAR (L) = VAR (L) U VAR (LHS (EXPR))
}
// include the statement as a slice
// VAR (L) is the slice variable ‘V‘ stored in list ‘L‘ and RHS (EXPR) denotes the right side of the expression and LHS (EXPR)
denotes the left side of the expression and VAR (RHS (EXPR)) denotes variables in the right side of the expression and VAR (LHS
(EXPR)) denote the variables in the left side of the expression.
5.2.2. else
do not include the statement as a slice
5.3. if (n is an element of a conditional statement)
{
5.3.1. if ((VAR (L) € LHS (EXPR) ¦ ¦ (VAR (L) € RHS (EXPR))
{
F= F U n //Store n
F= F U Loop body statements // Include all statements inside the conditional loop in F
}
}
5.4. if (n is an element of input statement)
F= F U n // include statement as a slice
5.5. if (n is an element of initialization or declaration statement)
F= F U n // include statement as a partial slice
}
6. else
n= n + 1
7. Repeat steps 5…6 until all the program statements are covered or till the EOP is reached
end
```

In the algorithm 2 given above, the user selects the program for which the test sequence is to be generated. The slicing criterion is verified initially. Slicing criterion contains the variable and statement number. Here, we have to check for the program statements that are affected by the value of a particular variable at a particular point. The slice variable ‘V‘ is stored in a list ‘L‘. The statement number is denoted by ‘n‘. The process starts from the (nth) line till the end the program is reached. In the (nth) line, it is checked whether the variable ‘V‘ is present or not. If the variable ‘V‘ is not present, then (n+1) th line is checked. If the variable ‘V‘ is present in the (n) the line, a series of steps are to be performed. If ‘V‘ is present in an expression, it is checked whether ‘V‘ is present on the right side or left side of the expression. If ‘V‘ is on the left side of the expression, that statement is considered as a slice and all the variables in the right side of the expression are also added to the list. In ‘V‘ is in the right side then it is not included as a slice. While checking the next line, we have to check not only for ‘V‘, but also all the all the variables present in the list. This is because; the other variables added to the list are the dependent variables of ‘V‘. Similarly, it is checked whether the slice variable is an element of conditional statement, declaration statement, input statement and output statement. If these conditions are true, the statements are considered as a slice. The statements inside the conditional body loop are also included as slice because the executions of these statements are dependent on the conditional clause. The process is repeated unit the end of the program and the result will be the forward slice for the corresponding

ii. System Description

An overview of our system model is given below. Our system is implemented using Java and Netbeans IDE. Netbeans is having extensible plug-in system and Java is having object-oriented features. This is why they have been used. The main modules of the implemented system consist of the following parts, given in figure 7.

1. Input unit
2. Slicer
3. Analyzer and tester

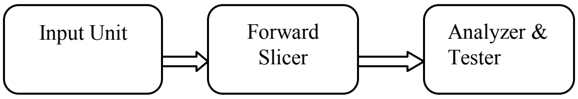


Figure 7: Main modules of slicing based system

a. Input unit

The input unit has the facility to select the software program which is to be tested. After selecting the program, the variables in the program are listed. From the listed variables, the user can select the variables for slicing criterion.

b. Forward Slicer

This is the main part of the system. In this unit, slicing is performed for the program which is to be tested. After getting the program and the list of variables from the input unit, forward slicing is performed to identify the relevant statements in the selected program with respect to the slicing criterion. Forward slicing is performed according to algorithm 2 given in section 3.3.1. A sample program code is given in Sample 1 and the working of forward slicing algorithm is explained below. In the program code given above in Sample 1, forward slicing is applied with respect to the input variable ‘basic‘. The slicing criterion given is C= (3, basic). The result of forward slicing is given in Result 1.

Sample 1

```
1. main( )
2. {
3. float basic, total, da, rent;
4. if (basic < 1000)
5. {
6. rent= basic * 12 /100;
7. da= basic * 60 / 100;
8.}
9. else
10. {
11. rent= 700;
12. da= basic * 80 / 100;
13.}
14. total =basic + rent + da;
15. System .out. println (—total = —+ total);
16.}
```

Result 1

```
4. if (basic < 1000)
6. rent= basic * 12 /100;
7. da= basic * 60 / 100;
9. else
11. rent= 700;
12. da= basic * 80 / 100;
14. total =basic + rent + da;
15. System .out. println (—total = —+ total);
```

The slicer will analyze the statements 4- 16 in Sample 1. Here statements 4, 6, 7, 9, 11, 12, 14, 15 will execute based on the value substituted for the variable ‘basic‘. We can notice that the dependencies are checked in a forward direction. The final value of variables ‘rent‘, ‘da‘ and ‘total‘ are dependent on ‘basic‘. Thus forward slices obtained can find if any errors are present in the dependent statements also. The resultant statements from forward slicing are given in Result 1.

c. Analyzer & Tester

In this unit the forward slices obtained are verified to find out whether they are significant in testing or not. Among the forward slices given above in Result 1, these statements are relevant in testing.

Testing using Slicing

```
4. if (basic < 1000)
6. rent= basic * 12 /100;
7. da= basic * 60 / 100;
9. else
11. rent= 700;
12. da= basic * 80 / 100;
14. total =basic + rent + da;
15. System .out. println (—total = —+ total);
```

The execution of the rest of the program statements is dependent on the value of the variable ‘basic‘. Here the tester identifies the test sequence statements which are relevant for generating the required test data values from the forward slices. In order to find the possible value of ‘basic‘ present in the conditional statement of the static slice, an equivalence partition method is applied. Equivalence partition is considered as the basis of all testing data generation methods and in this method, when a program works for a particular value in a partition, it may work for the other values in the same partition and this in turn helps to avoid duplicate testing [31]. Moreover, equivalence partition method is comparatively easy and reliable [31]. In equivalence partition, the input domain is divided into a number of sub domains. The sub domains make up the equivalence class. If a test data value in a class or partition is considered as a right value, then all the values under that particular class is considered as good values. We have to generate a value for the variable ‘basic‘ using equivalence partition. From the slice given in this section, conditional constraint is given is ‘if (basic<1000)‘. Here the possible partitions are (basic >1000)‘. and (basic >1000)‘. Using these partitions values are generated, which are given in table 11.

Table 11 : Valid and invalid test data values

Partition	Test Values	Result
> 1000	1500 1010 2000	Invalid
< 1000	800 900 700	Valid

From table 11, some of the valid and invalid test data values for the clause 'if (basic < 1000)' is obtained. The invalid test values is applicable to the 'else' part of the conditional clause 'if (basic < 1000)'. Substituting some of the test data values of 'basic' in the expressions will give the value of 'da' and 'rent' and finally the value of 'total' may be calculated from these data.

IV. Research Approach

In the previous sections, we have analysed program testing using genetic algorithm and program slicing methods. Some issues related to genetic algorithm based testing have also been pointed out. Based on these observations, we have framed some research questions (Q) in the coming section. The aims of the research questions are also mentioned and this may help future research work in this area.

a) *Research Questions*

Q1. What is the future of genetic algorithm based software testing?

The aim of this research question is to analyze the effectiveness of genetic algorithm based software testing. This question also intends to deal with the practical difficulties of this type of testing.

Q2. In the software testing context, why is program slicing considered a better approach?

This question aims to analyze the strengths of program slicing in testing and to study how program slicing makes testing more effective and reliable.

b) *Review Method*

We have referred to some relevant works in the field of genetic algorithm and program slicing based testing. A lot of works use genetic algorithms for test selection, test prioritisation, hardware testing etc. Apart from this, several works use a combined approach which uses genetic algorithm and other search algorithms for software testing [9]. Here we have mentioned only those works that describe software testing and test data generation using single population genetic algorithm. We have not considered other variations of genetic algorithms like parallel genetic algorithm as they are not employed in testing literature. We have reviewed several papers which describe slicing concepts, various types of slicing, slicing algorithms, applications of slicing etc. None of them have mentioned how to proceed to the testing phase after obtaining the slices. As our focus in on program slicing based software testing, we have selected some leading works which have mentioned the term 'testing' along with program slicing which is listed in table 10. Also, we have considered some of the fundamental works which use genetic algorithms for test case generation. We have not considered test selection, prioritisation etc. A summary of the referred works are given in table 5. The study made in section 3.2 answers the research questions.

V. Results

In this section we have tried to give an explanation to the research questions based on the studies mentioned in the previous sections.

Q1. What is the future of genetic algorithm based software testing?

We have provided only the most relevant points as solution to the research question. For this, the question Q1 has been split into some secondary questions (SQ). Providing appropriate answers to the secondary questions leads to an unbiased review of genetic algorithm based testing.

SQ1. What is the role of genetic operators in genetic algorithm based testing?

All the reviewed works use only single point crossover, except Jones et al. [30] work. In Jones's et al. [30] work, uniform crossover is used. Also, while others use simple mutation and Jones's work uses reciprocal and weighted mutation. Even though several works which explain the different types of operators and their relevance in different contexts exist, none of them have exploited these operators. They have used only the direct type of operators in their work. All these show that, the result obtained by using these common types of operators may be improved by substituting the testing process with a general operator selection strategy. This has not been decided till now in genetic algorithm based testing.

SQ2. Does population initialization and representation affect software testing?

From section 3.2, we can see that the population is selected randomly in most of the works. Selecting the population based on some heuristics improves the software testing process. Apart from this, we can see that only single population is used in most of these works. Only Wegner's et al. [52] work use multi-population along with single population. Even though a lot of research works are conducted continuously to decide the best type of population initialization, selection etc., some of the most common works which used genetic algorithm for software testing have experimented very little with population initialization methods. Again this shows that the quality of genetic algorithm based testing is dependent on population initialization and the lack of a general strategy for population setting makes the whole testing process unpredictable.

SQ3. What are the problems related with fitness function design during software testing?

Applying genetic algorithm in program testing requires optimizing the specified fitness function. A fitness function should be designed in such a way that it gives optimal solution for a given problem. Defining the fitness function imprecisely will lead to a wrong solution or in some cases the problem may get stuck in local optima [18, 25] suggested a method to remove variables which can lead to local optima. Even though, they were able to alleviate the problem of local optima, their approach didn't work for inner loop variables. Another problem faced during the fitness function design process is the dependency problem. While designing the fitness function for a target node, the dependent nodes which affect the target node should be considered. Since most of the works, which use genetic algorithm based approach for testing, do not use data flow criteria, the fitness value may not be correct. Some works were done on this area to minimize this problem, but they could not explain the best strategy for fitness function design in the context of testing [26, 50].

SQ4. Program dependency

In most of the genetic algorithm based software testing, program dependency is not correctly followed [37, 24]. In genetic algorithm based program testing, initially all the statements in the program should be analyzed to identify the relevant statements or we have to get the list of statements that will have a potential role in software testing. From the testing point of view, checking the whole program line by line is an unnecessary waste of effort. Instead of that, if we are able to find the program statements which help in program testing, such as those that assist in finding the test data values during testing, the whole testing effort will be reduced considerably. In addition, the testing can be made more methodical. Identifying the relevant statements which contribute to program testing, and analyzing those statements can give the dependence relation present in the program. Utilizing this dependence relation helps to trace out the errors in a program. For example, consider the sample control flow graph given below in Figure 8. All the program statements will be checked line by line from the starting point of the program. The statement 'basic<1000' assist in test data generation and suitable test data values should be generated for the variable basic'. The value of 'basic' is found out by optimizing the function f(x) = 1000 – basic. After finding out suitable values for the variable 'basic', the successive statements in the program is checked for errors. This is how the testing proceeds in this approach. In order to get a full satisfactory explanation for SQ4, we have to see the result research question Q2. The explanation given in Q2 provides a justification for SQ4.

Q2. In the software testing context, why is program slicing considered a better approach?

In the above section we saw some of the shortcomings of genetic algorithm based testing approach. An example given below gives an explanation to research question Q2. Consider the same example given in figure 8. In the control flow graph, the statements which correspond to each node are marked. From the control flow graph we are taking the forward slicing criterion as (2, basic). This means that all the statements which are affected by declaring the variable 'basic' in statement 2 is to be identified. The resultant nodes in the CFG are given below in Figure 9.

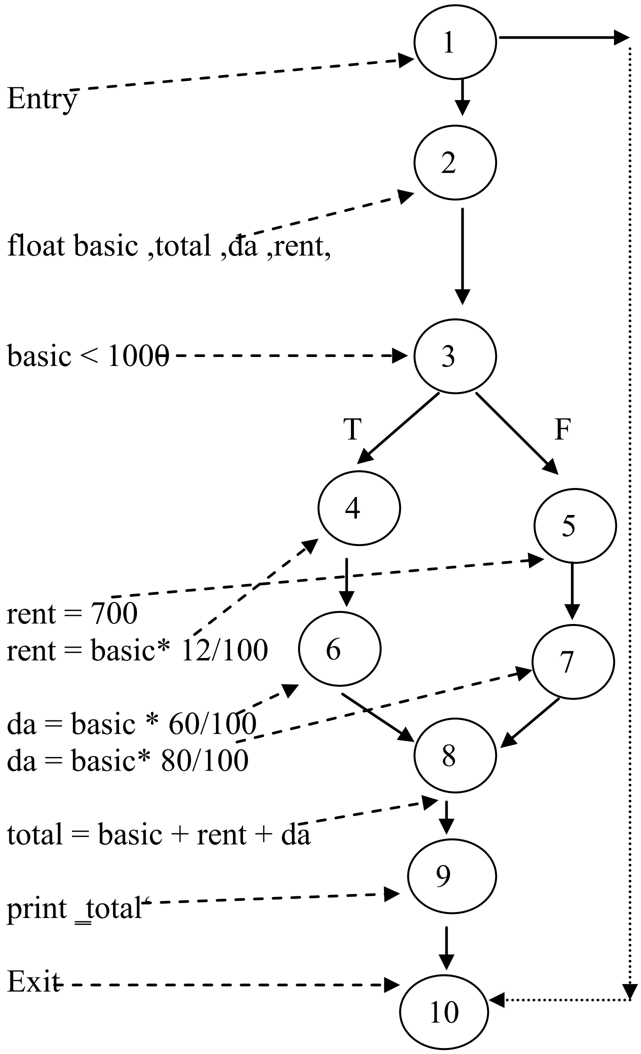


Figure 8 : Sample Control flow graph

It can be observed that all nodes displayed above will be affected by the variable ʻbasicʻ in statement 2. Node 3 is given as (basic<1000). When this program is to be tested, the test data which satisfies the condition in node 3 is to be generated. Similarly, nodes 4 and 6 are dependent on node 3 and this can be clearly traced form the slices obtained. Nodes 5 and 7 are also dependent on the variable ʻbasicʻ. If the value of ʻbasicʻ is greater than 1000, then these nodes get executed. From this we can conclude that the statements which are relevant in testing and in the successive stages of testing like test case generation can be identified easily by the process of slicing. Moreover, as slicing gives the dependence information present in a program, it will be easy to dig up the mistakes in the dependent statements.

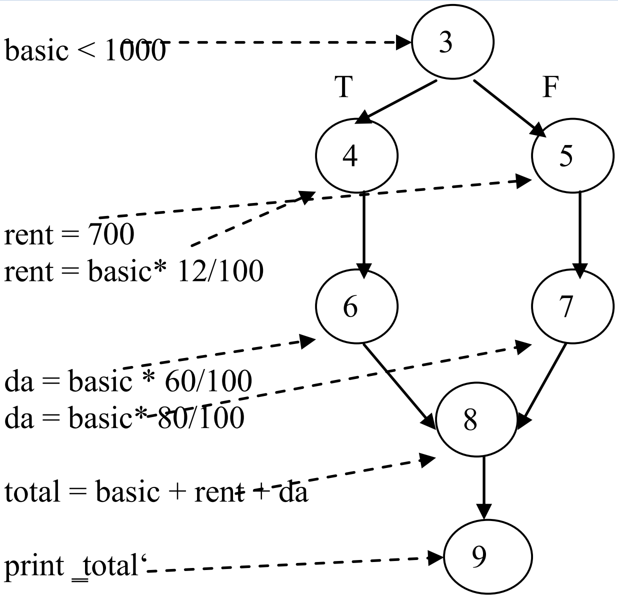


Figure 9 : Control flow graph obtained for slicing criterion (2, basic)

We saw that, for testing the same program given in figure 8, if genetic algorithm is used instead of program slicing, the program statements will be checked line by line from the starting point of the program. The main difficulty in this approach is that all the statements which contain relevant and irrelevant variables should be analyzed to trace the errors in the program code. On the other hand, as program slicing is done based on some slicing criterion, an overview of the dependence in the program code is revealed and error detection will be much easier. Here we can notice that every input variable present in a program will not be responsible for the execution of branches present in the program. Moreover, removing the irrelevant variables from a program and focusing only on the relevant variables which are significant in the execution of a target branch can improve the performance of genetic algorithm based testing. Relevant variables are those which can influence certain statements in a program, while irrelevant variables are those that cannot affect the program statements. This points out the fact that, genetic algorithm may not perform up to the mark in a practical program testing scenario [39], which underscores the superiority of program slicing in program testing. A graph is given in Figure 10 which gives an analysis of the performance of evolutionary algorithms with and without irrelevant variable removal. Here in y-axis the success rate is plotted and in x-axis the program names with branches are plotted. Here P1 denotes the program name, F1 denotes the function and B1, B2 and B3 denote different branches. Success rate is a measure of optimal test cases found out for the program branches. It can be noticed that the performance is better when irrelevant variables are removed from a program, compared to the performance without irrelevant variable removal. This establishes the weakness of genetic algorithm when there are a large number of irrelevant variables.

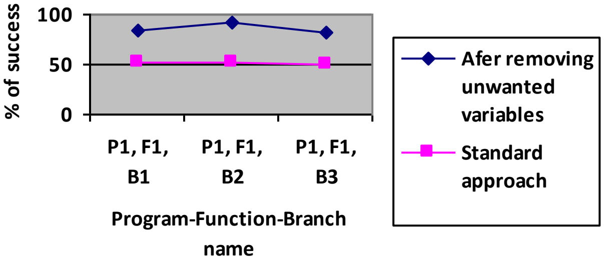


Figure 10 : Performance of evolutionary algorithms with and without irrelevant variable removal

Our observations, which are listed below, add more weight to the research question Q2. We have done an analysis of the number of program statements which have a significant role in program testing identified by both program slicing based testing and genetic algorithm based testing. The statements in a program have been considered as a metric for analyzing both these approaches to program testing. For a given program which is to be tested, the forward slicing covers more number of program statements compared to genetic algorithm in the same time span with respect to a particular variable. As the probability of error distribution in a program is uniform throughout the code, an increase in the number of executable statements with respect to a particular program variable increases the chance of discovering the number of faults related to that variable [33]. This means that, rather than concentrating on a particular area for a long time to attain high coverage for that particular branch or program code, program slicing tries to analyze more number of potential statements in a given program. Here the main principle is to identify possible program statements due to which program malfunctioning is caused, using minimal testing. This re-affirms the fact that program slicing can be more effective in program testing compared to genetic algorithm.

An assessment of testing productivity obtained in genetic algorithm and program slicing based testing approach is given in figure 11.

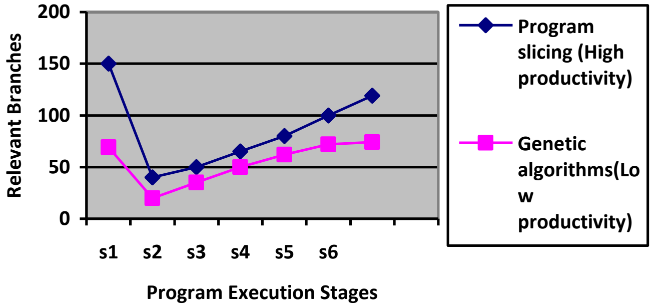


Figure 11 : Productivity graph

The graph shows that, when program testing is done using program slicing, there will be high testing productivity and when program testing is implemented using genetic algorithms, the testing productivity will be low. Some of the terms related to the graph in Figure 11 are given below.

1. Relevant branch indicates the statements of a program which may play a critical role in program testing.
2. Testing productivity indicates the measure of the number of relevant statements that can be covered in a specific time interval [31, 4].
3. High testing productivity means that more errors can be detected with less ‗effort‘, while low testing productivity means that the number of relevant statements covered in a specific time interval will be very few [31,4].
4. ‗Effort‘ means the time taken to detect the potential statements which contribute in program test data generation, run the program with the generated test cases and add the test cases to the test suit.

In program testing, the main objective is to find the maximum number of errors in the minimum time duration. Program slicing identifies more number of errors in less amount of time during the initial program execution stage. The relevant statements identified by program slicing provide an overview of dependency present in the program, making the error detection more practical. From this it is clear that, in program slicing based testing, although it is not possible to cover all the potential statements useful for testing, a reasonable number of statements can be analyzed when compared to genetic algorithm based program testing.

VI. Threats To Validity

The main threat to the validity of our work may be due to the limitation in the number and scope of the works which we have referred. We have limited our analysis to only those works which have mentioned the application of genetic algorithm in software testing and the use of program slicing in software testing.

The downside of such restriction in the selection of works was that, all the possible variants of genetic algorithm based testing have not been analysed. Also, we have not studied all the existing algorithms in program slicing which may have some relevance in the field of software testing. Our study has been limited to only those works which have explicitly mentioned the use of program slicing in testing. We feel that such a narrowing in the field of our study has sharpened its focus and enabled us to do an in depth analysis of our chosen study objectives; which being the identification of shortcomings of genetic algorithm and establishing the usefulness of program slicing in practical software testing.

VII. Conclusions

The unresolved issues in practical software testing constitute the Achilles‘ heel of software industry. As genetic algorithm is one of the most widely used and highly regarded approaches for software testing among researchers, it is high time that we explore its critical shortcomings in practical software testing. We have made an attempt to reveal some of the difficulties due to the inherent uncertain nature of genetic algorithm based software testing. A systematic review of the works made in this study reveals that, genetic algorithm factors like fitness function, population initialization and parameter settings impact the quality of solution obtained by genetic algorithm based testing. Apart from this, we have highlighted the significance of program slicing in software testing. For a given problem, program slicing has a higher 'testing productivity' with lesser 'effort'. We have used this principle as the nidus for developing our idea. We have put forth a forward slicing based method in this work. Checking of conditional constraints in the forward slices will help to pick out the rules which are to be fulfilled when testing is carried out. We have also discussed how the dependent statements in the slices are used to trace errors during testing. Certain analytical results are also provided in our work to substantiate these facts. With this work, we intend to provide a guide to future researchers and to make software industry aware of the scope and potential of using program slicing as an effective tool in software testing. In future, we plan to elaborate upon the issues brought forth by our work which may lead to promising developments in testing field.

References Références Referencias

1. AHMED, M. A. AND HERMADI, I. 2001. GA based Test Data Generator. *In the Proceedings of Congress on Evolutionary Computation (CEC'03)*. 1, 85-91.
2. AHMED, M. A. AND HERMADI I. 2007. GA based Multiple Paths Test Data Generator. *Computer and operations research*. 35, 3107-3124.
3. BATES, S. AND HORWITZ, S.1993. Incremental Program Testing using Program Dependence Graphs. *In the Proceedings of the 20th ACM Symposium on Principles of Programming Languages*.384-396.
4. BEIZER, B. 1990. *Software Testing Techniques,* Second Edition, International Thomson Computer Press, ISBN 1-85032-880-3.
5. BINKELY, D.1998. The Application of Program Slicing to Regression Testing. Loyola College in Maryland.1-24.
6. BLACK, R. 2007. *Pragmatic Software Testing: Become an Effective & Efficient Test Professional*. John Wiley& Sons Publishers.
7. BUENO, P. M. AND JINO, S. 2002. Automatic Test Data Generation for Program Paths Using Genetic Algorithms. *International Journal of Software Engineering and Knowledge Engineering,* 12, 6, 691-709.
8. CANFORA, G., CIMITILE, A. AND DE LUCIA, A. 1998. Conditioned Program Slicing. *Information and Software Technology,* 40, 11, 595–607.
9. CHEN, Y. AND ZHONG, Y. 2008. Automatic Path-oriented Test Data Generation Using a Multi-population Genetic Algorithm. *In the Proceedings of the Fourth International Conference on Natural Computation,* China, 566-570.
10. DEMILLI, R. A. AND OFFUTT, A. J. 1991. Constraint-Based Automatic Test Data Generation. *IEEE Transactions on Software Engineering,* 17,9, 900-910
11. DE LUCIA, A. 2001. Program Slicing: Methods and Applications. *In Proceedings of the 1st IEEE Workshop on Source code Analysis and Manipulation,* 142-149.
12. DOUNGSA-ARD, C., DAHA, K., HOSSAI, A., AND SUWANNASART, T. 2002. Test Data Generation from UML State Machine Diagrams using GAs. *In Proceedings of International Conference on Software Engineering Advances*.
13. FOX, C., DANICIC, S., HARMAN, M., AND HIERONS, R. M. 2004. Con SIT: a fully automated conditioned program slicer. *Software Practice and Experience,* 34, 15–46.
14. FOX, C., HARMAN, M., HIERONS, R. M., AND DANICIC, S. 2001. Backward Conditioning: A New Program Special is ation Technique and its Application to Program Comprehension. *In 9th IEEE International Workshop on Program Comprehension,* Los Alamitos, California, USA, 89–97.
15. GALLAGHER, K. B. AND BINKLEY, D. 2008. Program Slicing. *In the Proceedings of Frontier of Software Maintenance,* 58-67.
16. GALLAGHER, K. B. AND LYLE, J. R. 1991. Using Program Slicing in Software Maintenance. *IEEE Transactions on Software Engineering,* 17, 8 (Aug.), 751–761.
17. GALLAGHER, K. B. 1990. Using Program Slicing in Software Maintenance. *Ph.D. Thesis,* University of Maryland Baltimore County.
18. GOLDBERG, D. E. 1989. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison Wesley
19. GRAHAM-ROWE, D. 2002. Radio Emerges from the Electronic Soup. *New Scientist,* 175, 2358 (Aug), 19.
20. GUPTA, R., HARROLD, M.J., AND SOFFA, M. L. 1992. An Approach to Regression Testing using Slicing. *In the Proceedings of the IEEE Conference on Software Maintenance,* Orlando, Florida, 299-308.
21. HARMAN, M. AND DANICIC, S. 1994. Using Slicing to Simplify Testing. In the Proceedings of Eurostar.
22. HARMAN, M. AND BINKLEY, D. 2005. Forward Slices are Smaller than Backward Slices. In the Proceedings of the fifth International Workshop on Source code Analysis and Manipulation, 15-24.
23. HARMAN, M. AND DANICIC, S. 1998. A New Algorithm for Slicing Unstructured Programs. Journal of Software Maintenance and Evolution 10, 6, 415–441.
24. HARMAN, M., HASSOUN, Y., LAKHOTIA, K., MCMINN, P. AND J. WEGENER, J. 2007. The Impact of Input Domain Reduction on Search-based Test Data Generation. In *ESEC-FSE '07: Proceedings of the the 6th Joint Meeting of the*

European Software Engineering Conference and the ACM SIGSOFT Symposium on The Foundations of Software Engineering, New York, USA,155-164.

25. HARMAN, M., HU, L., ZHANG, X. AND MUNRO M. Side-effect Removal Transformation. 2001. *In the Proceedings of the 9th IEEE International Workshop on Program Comprehension,* Toronto, Canada, 310-319,
26. HARMAN, M., HU, L., HIERONS, R., BARESEL, A. AND STHAMER, H. 2002. Improving Evolutionary Testing by Flag Removal. *In Proceedings of the Genetic and Evolutionary Computation Conference,* New York, USA, 1359-1366.
27. Hill, T.A. 2002. Importance of Performance Stress Testing on Embedded Software Applications. *In the Proceedings of QA & Test Conference,* Spain
28. HOLLAND, J., H. 1975. *Adaptation in Natural and Artificial Systems,* University of Michigan Press, Ann Arbor.
29. HORWITZ, S., REPS, T. AND BINKLEY,D. 1988. Inter procedural Slicing using Dependence Graphs, *SIGPLAN Notices,* 23,7, 35–46.
30. JONES, B.F., STHAMER, H. H. AND EYRES, D. E. 1996. Automatic Structural Testing Using Genetic Algorithms. *Software Engineering Research Journal,* 299-306.
31. JORGENSEN, P.C.2008. *Software Testing: A Craftsman's Approach.* Auerbach Publications (Taylor and Francis group)
32. KOREL, B. 1990. Automated Software Test Data Generation, *IEEE Transactions on Software Engineering,* 16, 8,870-879.
33. KOREL, B. AND LASKI, J. 1988. Dynamic Program Slicing. *Information Processing Letters* 29, 3 (Oct.), 155–163.
34. KOREL, B. AND RILLING, J. 1998. Program Slicing in Understanding of Large Programs, In the Proceedings of the. 6th International Workshop on Program Comprehension, 145-152.
35. KOMONDOOR, R. AND HORWITZ, S. 2000. Semantics-preserving Procedure Extraction. In Proceedings of the 27th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages. ACM Press, N.Y., 155–169.
36. MANTERE, T. AND ALANDER, J. T. 2005. Evolutionary Software Engineering, A Review. *Journal of Applied Soft Computing,* 5,315-331.
37. MCMINN, P. 2004. Search-based Software Test Data Generation: A Survey. *Journal of Software Testing Verification and Reliability,* 14, 2, 105-156.
38. MCMINN, P., HARMAN, M., BINKLEY, D. AND TONELLA, P. 2006. The Species per Path Approach to Search-based Test Data Generation. In the Proceedings of the International Symposium on Software Testing and Analysis (ISSTA 06), Portland, Maine, USA, 13-24.
39. MCMINN, P., HARMAN, M., HASSOUN, Y., LAKHOTIA, K. AND WEGENER, J. 2012. Input Domain Reduction through Irrelevant Variable Removal and its Effect on Local, Global and Hybrid Search-Based Structural Test Data Generation, IEEE transactions on Software Engineering, 38, 2, 453-477.
40. MICHAEL, C. C., MCGRAW, G. E. AND SCHATZ M. A. 2001. Generating Software Test Data by Evolution. *IEEE Transactions on Software Engineering,* 27, 12 (Dec), 1085-1110.
41. MYERS, G. J. 1979. *The Art of Software Testing,* Wiley, New York.
42. OTT, L. M. AND BIEMAN, J. M. 1998. Program Slices as an Abstraction for Cohesion Measurement. *Information and Software Technology-Special issue on Program Slicing,* 40,691–700.
43. OUTT, A.J., Z. JIN, Z. AND PAN, J.1999. The Dynamic Domain Reduction approach to Test Data Generation. *Software Practice and Experience,* 29, 2, 167-193.
44. PARGAS, R. P, HARROLD, M. J. AND PECK, R. R. 1999. Test Data Generation Using Genetic Algorithms. *Journal of Software Testing, Verifications, and Reliability,* 9, 263-282.
45. PEI, M., GOODMAN, E. D, GAO, Z. AND ZHONG, K. 1994. Automated Software Test Data Generation Using A Genetic Algorithm. *Technical Report* , GARAG e of Michigan State University.
46. ROPER, M., MACLEAN, I., BROOKS, A., MILLER, J. AND WOOD, M. 1995. Genetic Algorithms and the Automatic Generation of Test Data. *Technical report* RR/95/195[EFoCS-19-95], Department of Computer Science, University of Strathclyde.
47. SAMUEL, P. AND MALL, R. 2009. Slicing based Test Case Generation from UML Activity Diagrams, ACM SIGSOFT *Software Engineering notices,* 34, 6, 1-14.
48. STHAMER, H. H. 1996. Automatic generation of Software Test Data using Genetic Algorithms, *Ph.D. Thesis,* University of Glamorgan, Pontyprid, Wales, Great Britain.
49. TIP, F. 1995. A Survey of Program Slicing Techniques, *Journal of Programming Languages,* 3, 3, 121-189.
50. TRACEY, N. 2000. A Search-Based Automated Test Data Generation Framework for Safety Critical Software. *Ph. D. thesis,* University of York.
51. WATKINS, A. 1995. A Tool for the Automatic Generation of Test Data using Genetic Algorithms. *In Proceedings of the fourth Software Quality Conference,* Dundee, Great Britain, 300-309.
52. WEGENER, J., BARESEL, A. AND STHAMER, H. 2001. Evolutionary Test Environment for Automatic Structural Testing. Journal of Information and Software Technology, 43, 841-854.

53. WEISER, M. Program Slicing. 1984. *IEEE Transactions on Software Engineering*. 10,4,352-357.

54. XANTHAKIS, S., ELLIS, C., SKOURLAS, C., LE GALL, A., KASTISKAS, S. AND KARAPOULIOS, K. 1992. Application of Genetic Algorithms to Software Testing. *In Proceedings of the 5th International Conference on Software Engineering and its Applications.*France,625-636.

This page is intentionally left blank

Global Journal of Computer Science and Technology: C
Software & Data Engineering
Volume 14 Issue 8 Version 1.0 Year 2014
Type: Double Blind Peer Reviewed International Research Journal
Publisher: Global Journals Inc. (USA)
Online ISSN: 0975-4172 & Print ISSN: 0975-4350

Optimization of Web-based Hierarchical Workgroups to Automate Workflow

By Jeba Moses .T, U. Palani & Parthiban .R

IFET College of Engineering, India

Abstract- This project is aimed at developing web application which provide services for an education institution such as requesting leave, reserving books and posting suggestions. It is an Intranet based web-application which can be accessed by all the students and faculties throughout the department. A different access level login has been provided for students, faculties and the head of the department to access the services. This application can be used to automate the workflow of leave applications and their approvals. Through this application students can send a leave request to both the head of the department and the faculty. Each and every leave request of the students will be stored in the database. Thus the students who took leave frequently can be identified easily by the head of the department. There are options available for the students and faculties to reserve the books from the department library.

A library catalogue has been provided with this application so that the desired books and magazines can be searched and reserved at anytime. After the reservation details are verified, the librarian or the managing staff will issue the books. In addition to this, department functions and upcoming events can be viewed through an event calendar.

Keywords: intranet, web app, library, service oriented.

GJCST-C Classification: D.3.4 G.1.6

Optimization of Web-based Hierarchical Workgroups to Automate Workflow

Jeba Moses .T [α], U. Palani [σ] & Parthiban .R [ρ]

Abstract- This project is aimed at developing web application which provide services for an education institution such as requesting leave, reserving books and posting suggestions. It is an Intranet based web-application which can be accessed by all the students and faculties throughout the department. A different access level login has been provided for students, faculties and the head of the department to access the services. This application can be used to automate the workflow of leave applications and their approvals. Through this application students can send a leave request to both the head of the department and the faculty. Each and every leave request of the students will be stored in the database. Thus the students who took leave frequently can be identified easily by the head of the department. There are options available for the students and faculties to reserve the books from the department library.

A library catalogue has been provided with this application so that the desired books and magazines can be searched and reserved at anytime. After the reservation details are verified, the librarian or the managing staff will issue the books. In addition to this, department functions and upcoming events can be viewed through an event calendar. Innovative thoughts and ideas are always accepted by the department, the Feedback or Suggestion Box scheme provided by this application gives an opportunity for the students to give their creative idea and getting them to be implemented for achieving department excellence

Index-terms: intranet, web app, library, service oriented.

I. Introduction

Automation is the process of converting a traditional system into a computer organization. Traditional system is no longer being dynamic and efficient as per our need. In order to overcome the defects of traditional system automation is introduced in our workflows. Some of the work flows which are going to be automated in this project are leave request, book reservation, suggestion box and an event calendar.

This web application is developed to optimize the system in a hierarchical environment. This application allows the user to access various automated services in an efficient and convenient manner.

Through this application students can able to send a leave request to their faculty and can reserve the book from the library then they can view the upcoming events and functions through the event calendar and can post their innovative ideas and thoughts through suggestion box.

This web application implements the automation in our workflow and it eliminates the manual process. Each and every data will be stored in the database so that the chances of losing data will be less.

a) *Existing System*

Existing systems are managed manually. Maintenance of data is very complex task. It requires lot of time to record or retrieve the details. The employee who has to record the details must perform their job very carefully. Even a small mistake would create many problems. All the operations must be performed in a perfect manner without any degradation.

b) *Disadvantages to Overcome*

- Loss of Data: A lot of paper works are needed for the safe keeping of the details.
- Time Wasting: user time is wasted as a result of searching for a book that has been borrowed by a user whose record cannot be traced on the paper records.
- Error Prone: The existing system of operation is prone to error.
- Tedious: It is tedious because it must take a routine
- Processing Speed: The processing speed is very low resulting into low output.

c) *Literature Survey*

i. *Leave Management System*

- In MNC's LMS is carried out by means of smart cards.
- Employees use their smart cards for login in order to access the services.
- After login they can request their leave to their higher officials.
- Each and every request of the employees will be stored in the form of database.
- Storing and retrieving of datas will be simple and the data loss has been greatly reduced.

ii. *Book Reservation System*

Before the advent of computer in modern age there are different methods of keeping records in the library. Records are kept in the library on shelves and each shelf are labeled in an alphabetical or numerical order, in which the categories of books available are

Author α: Dept of Information Technology, IFET College of Engineering, e-mail: jebamosesmtech@gmail.com
Author σ: Dept of Information Technology, IFET College of Engineering, e-mail: palani_uin@yahoo.com
Author ρ: Dept of Information Technology, IFET College of Engineering, e-mail: parthi_neyveli@yahoo.co.in

arranged on different position on the shelves and as well are recorded on the library manuscript and when any book is to be referenced the manuscript is being referred to, to know the position of such required book by the person that requested for the book.

After the invention of computer different researchers have carried out various approach on an automated library management system in which this project is as well all about.

The first library management system to be reviewed is the KOHA library management system. Since the original implementation in 1999, Koha functionality has been adopted by thousands of libraries worldwide, each adding features and functions, deepening the capability of the system.

The major setback of this library management system is that it is a web based and as a result it is not security conscious because hackers could have the database hacked and access or modify the information of such user.

Another Library Management System is the Capita's library software with the following benefits Increases support available for staff and users in any modern library service , Integrated, innovative system saves your library time, Improves the user experience.

The setback of this library management system is the cost of purchase and information generated from the software cannot be easily exported to be used in another system in case there is a system failure.

The set of researcher to be reviewed on an automated library system is a project carried out by Bhupendra, Singh Baghela, Shraddha Panwar, Vijay Vaishnav during as a partial fulfillment of the requirement for the System Design Project of Masters of Computer Application IV Semester, of the Rajasthan Technical University, Kota. The purpose of the application is for automation of library management.

II. Proposed System

Proposed system solves the inconvenience of existing system it is fully automated. In this system chance of losing data is not possible because data is maintained in the form of database. It is more flexible to access the information and also it will reduce the workload.

a) Advantages of the Proposed System

- Time consumption is less.
- Access control is provided.
- Reliability can be improved with the help of security.
- Give accurate results.

b) List of Modules

1. Registration, Login.
2. Leave request.
3. Book reservation.
4. Event calendar.
5. Suggestion box.

c) Module Description

i. *Registration*

This module is to register the details of the user. A new user has to register first in order to access the services. During the registration process the user has to give the details such as their Role, Register number (or) Employee-ID, Name, Department, Year, Date of birth, Mail-ID, Password, Cell number, Address, City, State.

After the details are given it will be stored in the database. The server has to check whether the username is already exist or not. If the username is already exists the user has to perform the registration again. After the successful registration the user can perform the login to access the services.

ii. *Login*

This module allows the user to perform the login. The login form holds two fields they are username and password.

The Mail-ID of the user will be acted as the username. The user has to give the username and password as mentioned in the registration form. Then the username and password are verified by obtaining the details stored in the database. If the username and password matches then the user will be authenticated to access the services.

After the successful login, based on the role their corresponding module will be displayed.

iii. *Leave Request*

In this module students can request their leave to the faculty by filling the leave request form. This form holds the field such as their Faculty Name, Purpose of Leave, Number of Days and Duration of leave. All the fields are mandatory. The student has to give all the details. After the details given by the student they can send the request. Leave request of the student will be send through mail.

Head of the department and the faculty have the rights to access the mail and send a response. Each and every request of the student will be stored in the database. Response to their request will be send by the faculty through notification. That notification will be send to the corresponding student. If the request of the student is in need of HOD's approval faculty can forward the request to the head of the department. Both the faculty and the Head of the department can have the rights to approve as well as decline the leave. The approval and cancellation of leave has to be updated in the database dynamically. There is also option available to search the leave details of a particular student and can also identify the student who took leave frequently

iv. *Book Reservation*

This module allows the student and faculties to reserve the book from the library. For that purpose a catalogue is provided which holds the details of books and magazines. If the user is in need of books it is very difficult to search the books manually. By using this module they can make an easy search by simply entering the name of the book in the search field. After the book details are entered the details about the book will be listed out. If the book is available they can reserve it otherwise they can search other book or logout from the application.

If the book is reserved the reservation details are send to the library staff they can verify the details and can issue the books to the corresponding user. Each and every reservation details are stored in the database. The user can just view the details of the book and its availability. The administrator can have the rights to manipulate the details. This module is designed to make the book search easy.

v. *Event Calendar*

This module can be accessed by all the registered users it allows the user to view information about the events in a chronological fashion. In the event calendar the upcoming events and functions are posted by the administrator. The events are highlighted in varied colors so that it is easy for the users to view the events easily. The administrator can have the rights to make modification like updation and deletion of events the user can just view it.

vi. *Suggestion Box*

This module is to post the suggestions, thoughts and ideas of the user through suggestion box. All the registered users can post their suggestions. It gives an opportunity for the users to post their creative thoughts and getting them to be implemented for department excellence.

III. System Architecture

a) *Registration and Login*

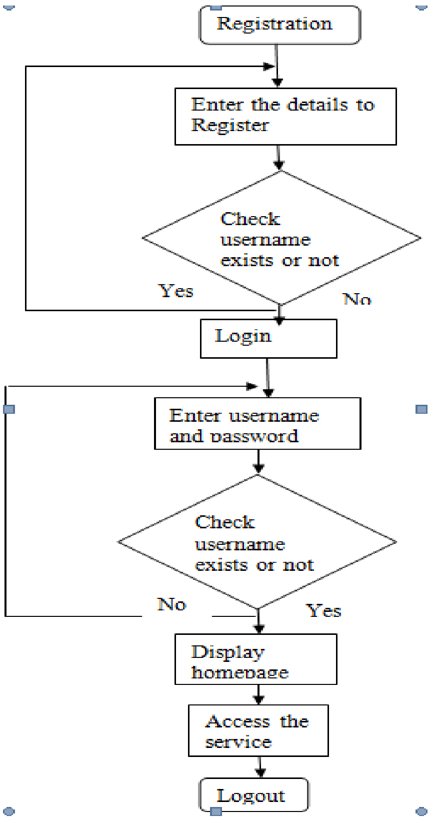

Registration
Enter the details to Register
Check username exists or not
Yes
No
Login
Enter username and password
Check username exists or not
No
Yes
Display homepage
Access the service
Logout

Figure 3.1 : Registration and Login

b) *Leave Request*

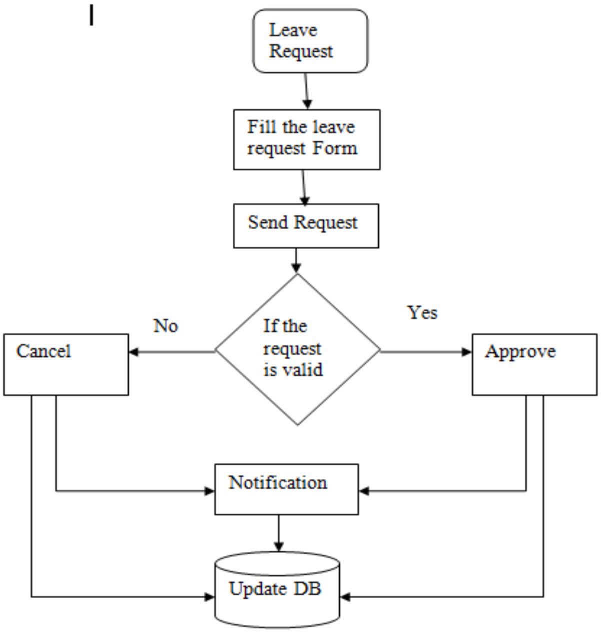


Figure 3.2 : Leave Request

c) Book Reservation

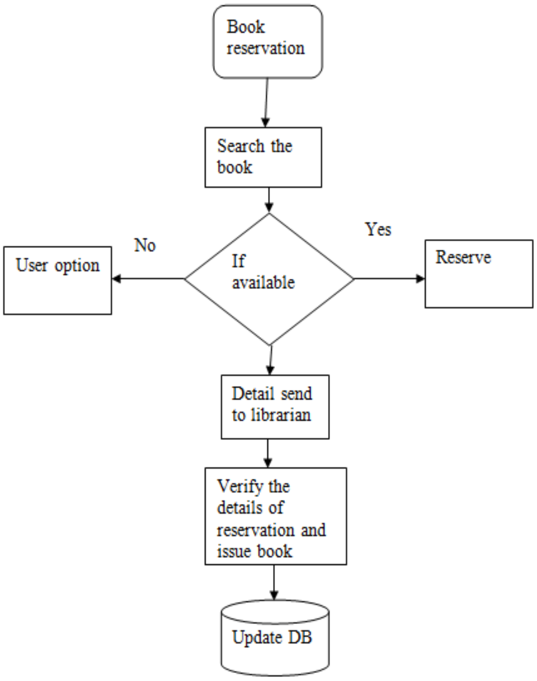


Figure 3.3 : Book Reservation

IV. SYTEM TESTING

System testing of software or hardware is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. System testing falls within the scope of black box testing, and as such, should require no knowledge of the inner design of the code or logic

a) Testing of Registration and Login

Table 4.1 : Testing of Registration and Login

S.NO	TESTCASE NAME	TESTCASE DESCRIPTION	PRECONDITION STEP	PASS/FAIL
1	REGISTRATION	New user have to fill the details to perform the registration	Fields are filled	Pass
			Fields are not filled	Fail
2.	LOGIN	Enter Username and Password as given in registration	Username entered without password	Fail
			Password entered without username	Fail
			Both Username and Password entered	Pass

b) Testing of Leave Request

Table 4. 2 : Testing of Leave Request

S.NO	TESTCASE NAME	TESTCASE DESCRIPTION	PRECONDITION STEP	PASS/FAIL
1	LEAVE REQUEST	Students have to select their faculty and fill the leave request form to send request	Fields are filled	Pass
			Fields are not filled	Fail

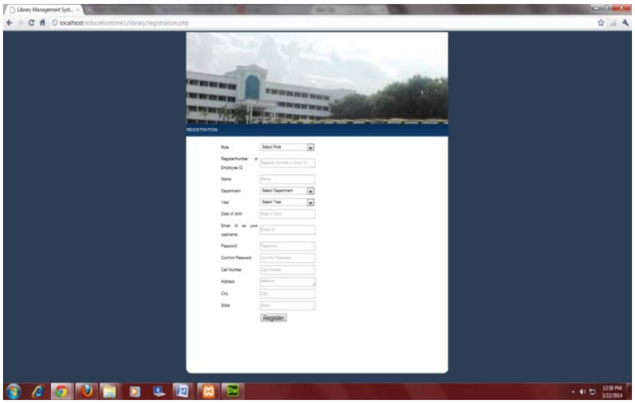
c) Testing of Book Reservation

Table 4.3 : Testing of Book Reservation

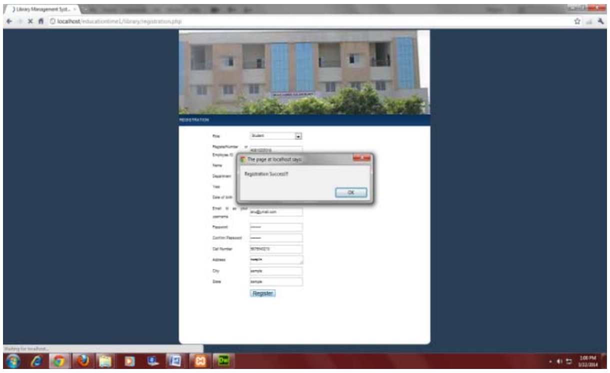
S.NO	TESTCASE NAME	TESTCASE DESCRIPTION	PRECONDITION STEP	PASS/FAIL
1	BOOK RESERVATION	Users can make a search of books	If book is available user can reserve the book	Pass
			If book is unavailable user can't reserve	Fail

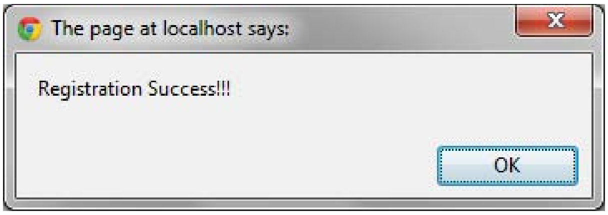
V. RESULT AND DISCUSSION

a) Registration



Registration success




The page at localhost says:

Registration Success!!!

Figure 5.1 : Registration

b) Login

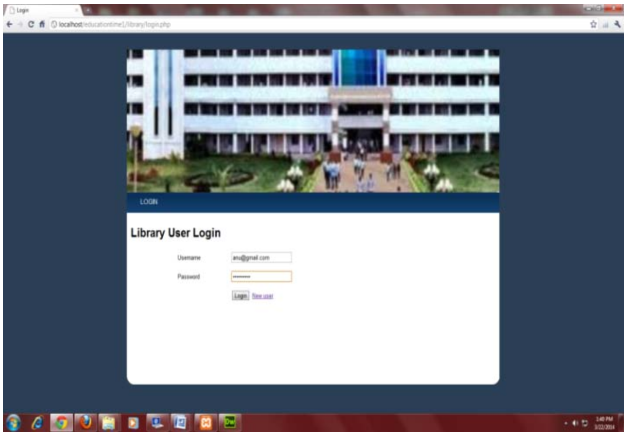
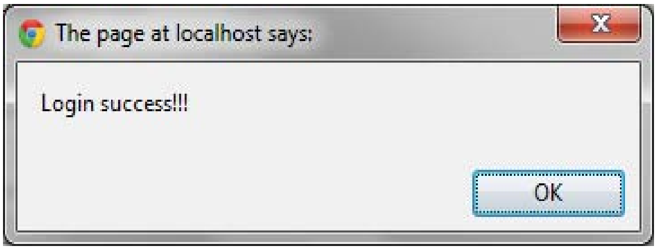


Figure 5.2 : Login

Login Success


The page at localhost says:

Login success!!!

c) Leave Request

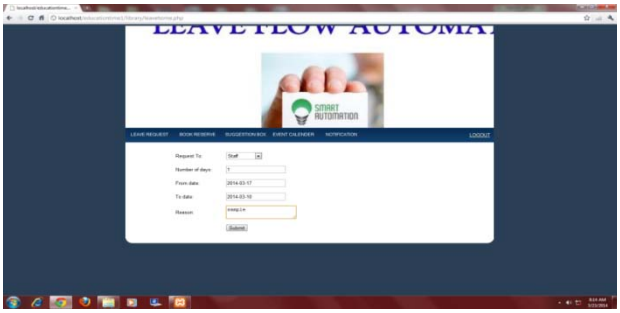
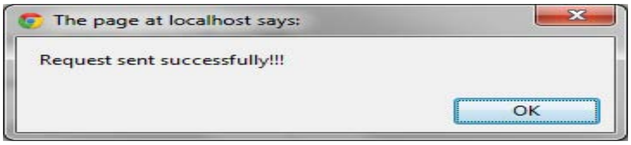


Figure 5.3 : Leave Request

Request Sent Successfully


The page at localhost says:

Request sent successfully!!!

d) Book Reservation

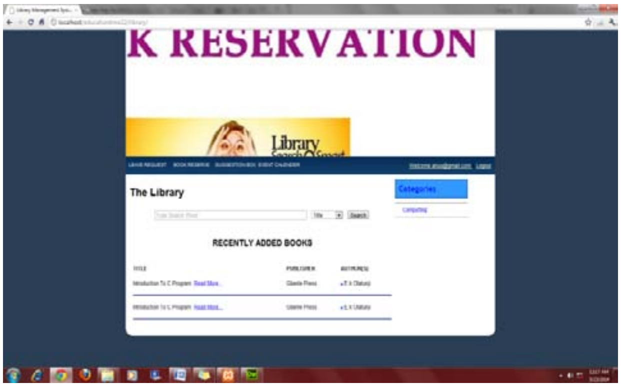


Figure 5.4 : Book Reservation

i. Category search

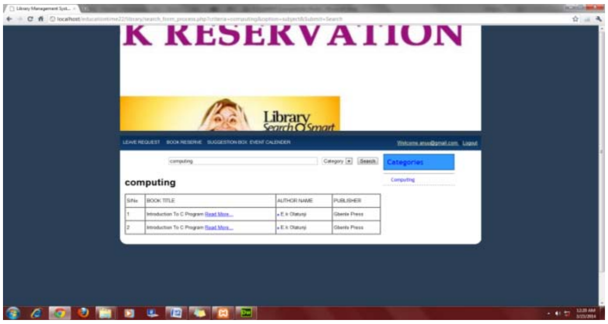
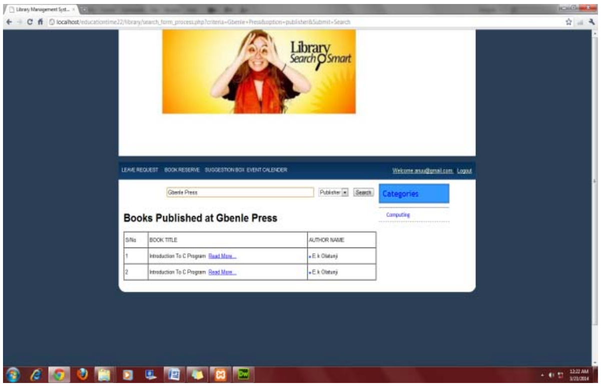


Figure 5.4.1 : Category Search

ii. *Publisher search*



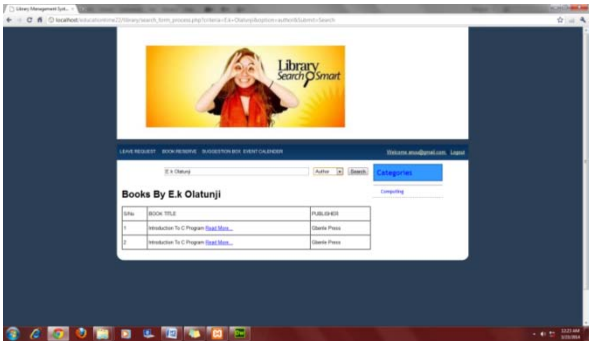
Library Search Smart
LEAVE REQUEST BOOK RESERVE SUGGESTION BOX EVENT CALENDER
Welcome anuu@gmail.com Logout
Gbenle Press
Publisher
Search
Categories
Computing

Books Published at Gbenle Press

S/No	BOOK TITLE	AUTHOR NAME
1	Introduction To C Program Read More...	E.k Olatunji
2	Introduction To C Program Read More...	E.k Olatunji

Figure 5.4.2 : Publisher Search

iii. *Author search*



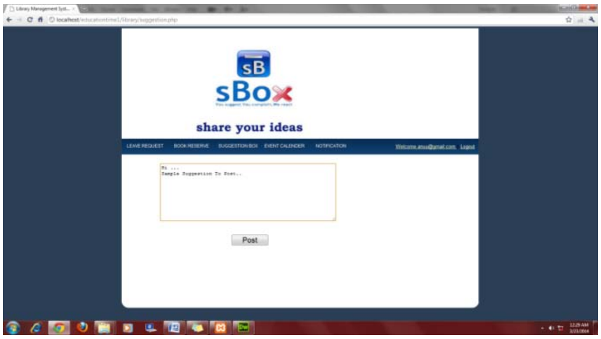
Library Search Smart
LEAVE REQUEST BOOK RESERVE SUGGESTION BOX EVENT CALENDER
Welcome anuu@gmail.com Logout
E.k Olatunji
Author
Search
Categories
Computing

Books By E.k Olatunji

S/No	BOOK TITLE	PUBLISHER
1	Introduction To C Program Read More...	Gbenle Press
2	Introduction To C Program Read More...	Gbenle Press

Figure 5.4.3 : Author Search

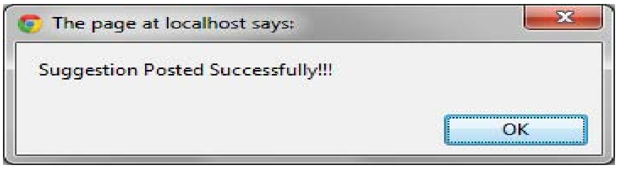
e) *Suggestion Box*



sB
sBox
share your ideas
LEAVE REQUEST BOOK RESERVE SUGGESTION BOX EVENT CALENDER NOTIFICATION
Welcome anuu@gmail.com Logout
Post

Hi ...
Sample Suggestion To Post..

Figure 5.5 : Suggestion Box



The page at localhost says:

Suggestion Posted Successfully!!!

f) *Event Calendar*

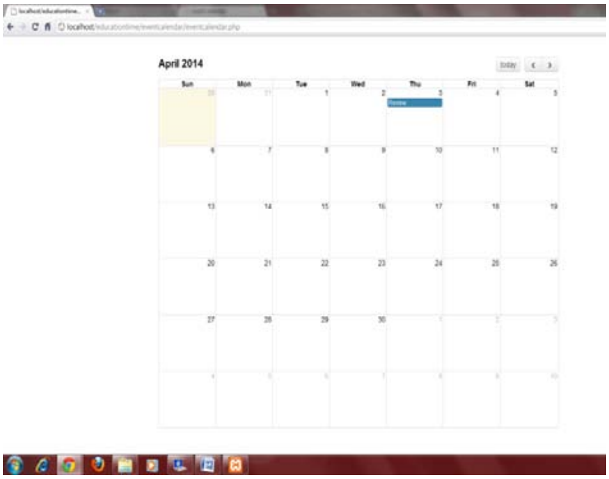


Figure 5.6 : Event Calendar

VI. Conclusion and Future Enhancement

The primary goal of the project is to automate the workflow in a hierarchical environment that has been implemented successfully with the help of Dreamweaver as a text editor, php as a scripting language and mysql as a back end database.

This application can provide a number of services which enhance the automated process.

The scope of the project is not having an end. This application is designed to automate the workflow for a single department. In future it will be developed to automate the workflow for a college.

References Références Referencias

1. www.w3schools.com
2. www.php.net
3. www.tutorialspoint.com
4. www.freeprogrammingresources.com/phpbook.html
5. www.freetechbooks.com

Global Journal of Computer Science and Technology: C
Software & Data Engineering
Volume 14 Issue 8 Version 1.0 Year 2014
Type: Double Blind Peer Reviewed International Research Journal
Publisher: Global Journals Inc. (USA)
Online ISSN: 0975-4172 & Print ISSN: 0975-4350

Assessing Weight of the Coupling between Objects towards Defect Forecasting in Object Oriented Programming

By A. Saidulu

Abstract- This study presents a new approach for predicting fault proneness in object classes. A new metric called weighted coupling between objects or wcbo, measures a new unit called weighted coupling between objects. The approach then ranks each class of the object based on its fault proneness using the HITS algorithm that measures hub and authority weights in link analysis. The proposed metrics advantage lies in the fact that it can be computed without using any programming language, in a simpler way, comparatively. The study undertook with huge open source systems to forecast fault proneness, shows the projected wcbo fault proneness sensitivity to be 98.7% and compared with CBO which was 42% sensitive.

Keywords: software engineering, design metrics, defect forecasting, coupling between objects, cohesion.

GJCST-C Classification : I.2.5 D.3

Assessing Weight of the Coupling between Objects towards Defect Forecasting in Object Oriented Programming

A. Saidulu

Abstract- This study presents a new approach for predicting fault proneness in object classes. A new metric called weighted coupling between objects or wcbo, measures a new unit called weighted coupling between objects. The approach then ranks each class of the object based on its fault proneness using the HITS algorithm that measures hub and authority weights in link analysis. The proposed metrics advantage lies in the fact that it can be computed without using any programming language, in a simpler way, comparatively. The study undertook with huge open source systems to forecast fault proneness, shows the projected wcbo fault proneness sensitivity to be 98.7% and compared with CBO which was 42% sensitive.

Keywords: software engineering, design metrics, defect forecasting, coupling between objects, cohesion.

I. Introduction

Software fault proneness and maintainability has been outlined imprecisely owing to the complexity and confines in assessing objects failure and reusability. Object-oriented designs two key concepts are coupling and cohesion, however evaluation of reusability of objects applying coupling and cohesion has been limited. Several metrics proposed tried to quantify coupling and cohesion to analyze software fault proneness and maintainability. The paper presents a statistical model with abstraction to contain the degree of fault proneness in OO systems using HITS link analysis algorithm [1].

Object oriented analysis and design primary objective is, software arrangement classes should allow maximum cohesion and low coupling. Coupling is the extent to which the assorted objects interact. Cohesion is the extent to which the functions executed by a subsystem are related. The class design gives an insight of activities, testing efforts, continuance tasks and reuse. Structural metrics coupling and accord are present in majority in structural information. They assess relations like attributes usages or adjustment calls and compute the degree of alternation and relationships in allocation of preceding code essentials such as classes, methods and attributes in acquisitive (OO) software systems. An object is accepted as error free and simpler to adjust if it is in form, calculated and its functionality has been appropriately disseminated to its several dependent objects.

However if the objects are exceptionally inter-reliant then the radical ones are expected to have cogent affects on the performance of others. Proper distribution of function thus forms the basis of the two OO design concepts i.e. coupling and cohesion.

If a reliant object is altered in levels of various functions, then the distribution of functionality is carried to reliant objects and the metrics, cohesion and coupling used to evaluate modified tasks like design quality [2], [3], collision analysis [5], [6], [8], naming the design patterns [11], forecasting software quality [7] and errors [8], [9], [10], etc are accepted advantageous. So for a calculated object in form, ascending cohesion is proper and loose coupling among dependent objects is a helpful characteristic of an object. However to address the issue of objects that that are neither calculated nor dependent, this paper proposes a metric called weighted coupling between objects or wcbo.

The remaining paper is organized as, related work in section II , proposed wcbo metric measuring process in section III, process explored with example in section IV, results and analysis presented in section V, concluding the proposal in section VI and references in section VII.

II. Related Work

The structural metrics analyzing the adeptness of the appulse weight of the anniversary chic with CBO gives inaccurate accountability decumbent coupled classes. In the best of our knowledge and from articles cited recently in conferences and journals, it is evident that the CBO and other CBO related metrics are not sensible to be considered as metrics to predict fault proneness.

The issue of predicting fault-prone classes or simply bug forecast in software has been discussed in seminars and journal publications, generated a number of research papers in the previous 10 years and is an on the go area of research. Unique techniques were designed such as PROMISE [12] and MSR [13] with their specific data sets for prediction of classes fault-prone in software. This paper defines new intangible

Author : M. Tech in Computer science. e-mail: siddu33@gmail.com.

metrics for class cohesion and coupling which is distinctive and is an enhancement over earlier cutting edge work. Current research demonstrates that software metrics can be used as good signs for the fault proneness of classes in OO systems [3], [8], [9], [10], [14], [15], [16], [17], [18]. Specific models available use machine learning [9] and logistic regression analysis [3], [9], [14], [15], [16], [17], [18] to develop metric-based models for predicting class faults. So a metric called weighted coupling between objects is presented. The weighted metric coupling assesses using an innovative statistical evaluating method governed by conditions of HITS algorithm [1]

III. Weighted Coupling between Objects (WCBO) Metric

a) *Hypothesis*

Coupling between objects being high or excessive [19] is unfavorable to modular design and inhibits object reusability [19] [20]. In order to improve modularity and promote encapsulation, inter-object class couples should be kept to a minimum. The more independent a class is, the easier it is to reuse it in another application [19] [21] [22]. As the number of couples increase the sensitivity to changes in other parts of the design automatically increases resulting in maintenance problems [23]. The fault proneness of a class is more if it has high coupling and also if it has more import coupling compared to export coupling [24]. Thus the requirement of rigorous testing arises.

Previous research states [24] the metric CBO compared to other metrics has high sensitivity in predicting fault proneness which however is the possibility predicting the fault proneness rather than predicting the fault proneness. CBO also ranks the objects by their fault proneness that is not sensible. Hence the metric weighted coupling between objects is proposed to predict the fault proneness with high sensitivity. The description of the WCBO measurement process is as follows.

b) *Assumptions*

Let's consider a set of classes $c1, c2, c3, \ldots\ldots., cn$

Two classes 'ci' and 'cj', where 'ci' is coupled with 'cj' if and only if, any of the methods of 'ci' invoke any of the methods that belongs to 'cj' [19]. A graph known as directed graph is created. Vertices represent classes' i.e. $c1, c2, c3, \ldots\ldots., cn$. Edges are flanked by classes and an edge is feasible if at least one method of source class invokes the method of the target class.

A set of classes where each class is shown by an individual path and joined by edges is viewed as a single connected transaction ct.

The vertices belonging to a set of classes in a connected transaction is taken as connected set cs. The set of all the connected sets is indicated as 'SCS'.

c) *Process*

To determine "wcs' of every class, first a bi parted graph is created between all feasible connected sets related to connected sets SCS and set of all classes.

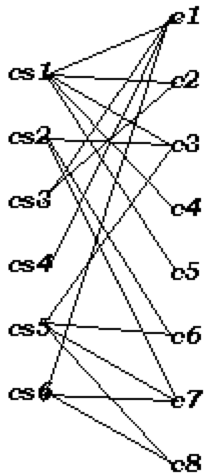


Figure 1 : Bi-partite graph between connected sets and classes

A set of connected sets SCS is shown as a bipartite graph with zero loss of information. Let $SCS = \{cs_1, cs_2, cs_3, \ldots., cs_m\}$ be a record of connected sets and $C = \{c_1, c_2, c_3, \ldots\ldots, c_n\}$ be the related set of classes. Evidently then is equivalent to the bipartite graph $G = (SCS, C, E)$ where $E = \{(cs, c) : c \in cs, cs \in SCS, c \in C\}$.

In Fig 1, the bipartite graph shows a class c that has coupling support proportional to its degree of fault proneness. Connection between connected sets and classes shown in the graph is similar to the association of hubs and authorities in the HITS model [14]. The relation between connected sets and classes is evaluated from their weights; however obtaining the weights from a set of connected sets is a challenge. Naturally a connected set having high coupling weights is supposed to contain several classes with high coupling support. Similarly a class having high coupling support is contained by many sets with high coupling weights. Having different coupling weights for different connection sets is decisive to show their different levels of importance and for working with link-based models for evaluation of connected sets with high coupling weights. The HITS algorithm can be applied to this bipartite graph considering the sets as "pure" hubs and the classes as 'pure' authorities and explore the process as below.

Matrix A represents connected sets and class connections as binary matrix (table 1) i.e. shows the connection between a class and each connected set .

Table 1 : Matrix A

	C1	C2	C3	C4	C5	C6	C7	C8
Cs1	1	1	1	1	1	0	0	0
Cs2	0	0	1	0	0	1	1	0
Cs3	1	1	0	0	0	0	0	0
Cs4	1	0	0	0	0	0	0	0
Cs5	0	0	1	0	0	1	1	1
Cs6	1	0	0	0	0	0	1	1

Value 1 represents that this class c is existing in related connected set cs

If $c \in cs$ then matrix value is 1

Else if $c \notin cs$ then matrix value is 0

Consider the matrix that represents each hub initial value as 1 (see fig 2).

1
1
1
1
1
1
1
1
1
1

Figure 2 : Each hub weight is considered as 1 by default and shown as a matrix as below

Transpose the matrix A as A'

Table 2 : Transpose matrix of matrix below shows the connection between a class and each connected set

	Cs1	Cs2	Cs3	Cs4	Cs5	Cs6
C1	1	0	1	1	0	1
C2	1	0	1	0	0	0
C3	1	1	0	0	1	0
C4	1	0	0	0	0	0
C5	1	0	0	0	0	0
C6	0	1	0	0	1	0
C7	0	1	0	0	1	1
C8	0	0	0	0	1	1

Authority weights are determined by multiplying A' with u i.e. $v = A' \times u$. Matrix representation of authority weights, matrix v is obtained by multiplying A' and u.

The original hub weights are obtained from matrix multiplication between $A and\ v$.

$$u = A \times v$$

Then the wcbo of class c can be measured as follows

$$wcs(c) = \frac{\sum_{i=1}^{m} \{u(cs_i) : c \in cs_i\}}{\sum_{i=1}^{m} u(cs_i)}$$

Then the wcbo of coupling between classes c_i and c_j can be measured as follows

$$wcs(c_j \to c_k) = \frac{\sum_{i=1}^{m} \{u(cs_i) : (c_j, c_k) \subset cs_i \wedge (c_j \to c_k) \in cs_i\}}{\sum_{i\ 1}^{m} u(cs_i)}$$

IV. Ranking Objects and Find Fault Prone Coupling using WCBO: An Example

Let's consider the bi-partite graph in figure 1, the Table 1 is the matrix A generated from that bi-partite graph and the Table 2 is transpose matrix A' of matrix A. Initial hub values:

$$u = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

The resultant matrix v generated from $v = A' \times u$ is

$$v = \begin{bmatrix} 4 \\ 2 \\ 3 \\ 1 \\ 1 \\ 2 \\ 3 \\ 2 \end{bmatrix}$$

Original hub values measured by $u = A \times v$ give results as;

In Table 4 each class is listed from their highest to lowest values of their wcbo values. The classification shows 'c_1' is highly fault prone.

Table 4 : Values of weight coupling between objects of the classes of Bi-Partite graph

Class	wcbo value
C1	0.625
C2	0.354
C3	0.604
C4	0.229
C5	0.229
C6	0.375
C7	0.562
C8	0.396

The coupling between 'c_1' to other classes listed in Table 5.

Table 5 : Weight coupling between objects of 'c_1' to other classes (export coupling)

$c_1 \to c_2$	0
$c_1 \to c_3$	0.229
$c_1 \to c_4$	0.229
$c_1 \to c_5$	0
$c_1 \to c_6$	0
$c_1 \to c_7$	0.187
$c_1 \to c_8$	0.187
$c_2 \to c_1$	0
$c_3 \to c_1$	0
$c_4 \to c_1$	0
$c_5 \to c_1$	0
$c_6 \to c_1$	0
$c_7 \to c_1$	0
$c_8 \to c_1$	0

In Table 5 wcbo connection of c_1 with other classes' shows, though c_1 is ranked high in fault analysis, c_1 fault proneness is confined to c_4, c_5, c_7, c_8.

The degree of fault proneness $dfp(c)$ for a class c can be measured as follows;

$$dfp(c_i) = \frac{\sum_{j=1}^{m} \{wcs(c_i \to c_j) + wcs(c_j \to c_i) : i \neq j\}}{wcs(c_i)}$$

In the case of class c_1, the degree of fault proneness is 1.3312.

V. Results Analysis

The tests performed ensuring high SDLC standards, on application classes with sets of diverse numbers, calculated the accuracy of fault proneness of wcs.

$$\text{S(wcs)} = \frac{\text{Classes correctly predicted as fault prone}}{\text{Classes actually fault prone}}$$

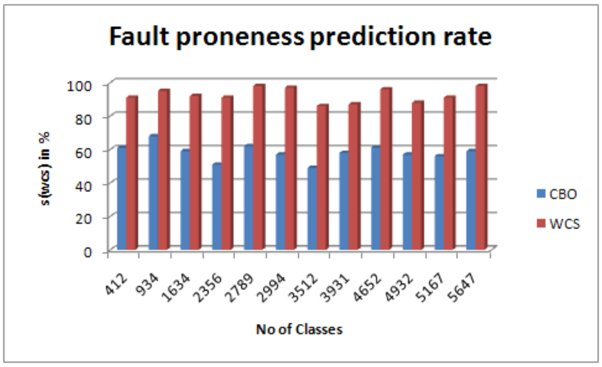


Figure 3 : percentage of Fault proneness prediction accuracy

Sahraoui, God in & Miceli [23] in their empirical studies showed CBO>14 is very high against the observed value in our test where "dfp " value higher than 4.65 is severity in fault proneness. In the above graph the weight coupling between objects effectively forecasts degree of fault proneness with 90% accuracy, very efficient compared to CBO.

VI. Conclusion

Our research in software engineering is on the ability to forecast the degree of fault proneness in OO systems. The study first evaluated previous research citing, CBO metric value is directly proportional to fault proneness, proved theoretical and lacked support. Thus a new metric is projected, namely weighted coupling between objects which measures the weighted coupling between objects of each class and related coupled classes. Further it was established in predicting the degree of fault proneness, an individual class weight coupling between objects of is insufficient, whereas the wcbo of classes with coupling is able to forecast with an accuracy of 91% and is highly efficient compared to CBO. Further research will focus on developing the statistical approach to calculate the weighted cohesion support and evolving the strategy for predicting the degree fault proneness.

References Références Referencias

1. J.M. Kleinberg, "Authoritative Sources in a Hyperlinked Environment," J. ACM, vol. 46, no. 5, pp. 604-632, 1999.
2. Bansiya, J. and Davis, C. G., "A hierarchical model for object-oriented design quality assessment", IEEE TSE, vol. 28, no. 1, pp. 4-17., 2002.
3. Briand, L. C., Wüst, J., Daly, J. W., and Porter, V. D., "Exploring the relationship between design measures and software quality in object-oriented systems", Journal of System and Software, vol. 51, no. 3, pp. 245-273., 2000.
4. Briand, L., Wust, J., and Louinis, H., "Using Coupling Measurement for Impact Analysis in OO Systems", in IEEE ICSM'99, pp. 475-482., 19996.
5. Poshyvanyk, D., Marcus, A., Ferenc, R., and Gyimóthy, T., "Using Information Retrieval based Coupling Measures for Impact Analysis", Empirical Software Engineering, vol. 14, no. 1, pp. 5-32., 2009.
6. Wilkie, F. G. and Kitchen ham, B. A., "Coupling measures and change ripples in C++ application software", JSS, vol. 52, pp. 157-164., 2000.
7. Lawrie, D., Feild, H., and Binkley, D., "Leveraged Quality Assessment Using Information Retrieval Techniques", in ICPC'06, pp. 149-158., 2006.
8. El-Emam, K. and Melo, K., "The Prediction of Faulty Classes Using Object-Oriented Design Metrics", NRC/ERB-1064, vol. NRC 43609, 1999.
9. Gyimóthy, T., Ferenc, R., and Siket, I., "Empirical validation of OOmetrics on open source software for fault prediction", TSE,vol.31/10,Oct' 2005.
10. Quah, T.-S. and Thwin, M. M. T., "Application of neural networks for software quality prediction using OO metrics", in ICSM'03,pp. 116-125.
11. Antoniol, G., Fiutem, R., and Cristoforetti, L., "Using Metrics to Identify Design Patterns in Object-Oriented Software", in Proc. of 5th IEEEMETRICS'98, Bethesda, MD, pp. 23 - 34., 1998.
12. http://promisedata.org/
13. http://msr.uwaterloo.ca/
14. Arisholm, E., Briand, L. C., and Foyen, A., "Dynamic coupling measurement for OO software", IEEE TSE, vol. 30, no. 8, pp. 491-506., 2004.
15. Basili, V. R., Briand, L. C., and Melo, W. L., "A Validation of OO Design Metrics as Quality Indicators", IEEE TSE, vol. 22/10, Oct., pp. 751-761., 1996.
16. Briand, L., Melo, W., and Wust, J., "Assessing the Applicability of Fault-Proneness Models across OO Software Projects", TSE, vol.28/7,706-720., 2002.
17. Olague, H., Etzkorn, L., Gholston, S., and Quattlebaum, S., "Empirical Validation of Three Software Metrics Suites to Predict Fault-Proneness of Object-Oriented Classes Developed Using Highly Iterative or Agile Software Development Processes", IEEE TSE, vol. 33, no. 6, pp. 402-419., 2007.
18. Subramanyam, R. and Krishnan, M. S., "Empirical Analysis of CK Metrics for Object-Oriented Design Complexity: Implications for Software Defects", IEEE TSE, vol. 29, no. 4, pp. 297-310., 2003.
19. Shyam R. Chidamber, Chris F. Kemerer. A Metrics suite for Object Oriented design. M.I.T. Sloan School of Management E53-315. 1993. http://uweb.txstate.edu/~mg43/CS5391/Papers/Metrics/OOMetrics.pdf
20. Lionel C. Briand, John W. Daly, and Jürgen Wüst: A Unified Framework for Coupling Measurement in Object-Oriented Systems. Fraunhofer Institute for Experimental Software Engineering. Kaiserslautern, Germany. 1996. http://www.iese.fraunhofer.de/network/ISERN/pub/ technical_reports/isern-96-14.pdf
21. Victor Basili, Lionel Briand and Walcelio Melo. A Validation of Object-Oriented Design Metrics as Quality Indicators. IEEE Transactions on Software Engineering. Vol. 22, No. 10, October 1996. http://www.cs.umd.edu/users/basili/publications/journals/J60.pdf
22. Roger Whitney: Course material. CS 696: Advanced OO. Docs 6 & 8, Metrics. Spring Semester, 1997. San Diego State University. http://www.eli.sdsu.edu/courses/spring97/cs696/notes/metrics/metrics.html and http://www.eli.sdsu.edu/courses/ spring97/cs696/notes/metrics2/metrics2.html
23. Houari A. Sahraoui, Robert God in, Thierry Miceli: Can Metrics Help Bridging the Gap Between the Improvement of OO Design Quality and Its Automation? http://www.iro.umontreal.ca/~sahraouh/papers/ICSM00.pdf
24. K.K. Aggarwal, Yogesh Singh, Arvinder Kaur, Ruchika Malhotra; Investigating effect of Design Metrics on Fault Proneness in Object-Oriented Systems; JOURNAL OF OBJECT TECHNOLOGY; Online at http://www.jot.fm. Published by ETH Zurich, Chair of Software Engineering ©JOT, 2007

Fellows

FELLOW OF ASSOCIATION OF RESEARCH SOCIETY IN COMPUTING (FARSC)

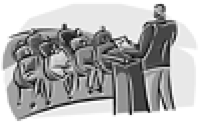
Global Journals Incorporate (USA) is accredited by Open Association of Research Society (OARS), U.S.A and in turn, awards "FARSC" title to individuals. The 'FARSC' title is accorded to a selected professional after the approval of the Editor-in-Chief/Editorial Board Members/Dean.

> The "FARSC" is a dignified title which is accorded to a person's name viz. Dr. John E. Hall, Ph.D., FARSC or William Walldroff, M.S., FARSC.

FARSC accrediting is an honor. It authenticates your research activities. After recognition as FARSC, you can add 'FARSC' title with your name as you use this recognition as additional suffix to your status. This will definitely enhance and add more value and repute to your name. You may use it on your professional Counseling Materials such as CV, Resume, and Visiting Card etc.

The following benefits can be availed by you only for next three years from the date of certification:

FARSC designated members are entitled to avail a 40% discount while publishing their research papers (of a single author) with Global Journals Incorporation (USA), if the same is accepted by Editorial Board/Peer Reviewers. If you are a main author or co-author in case of multiple authors, you will be entitled to avail discount of 10%.



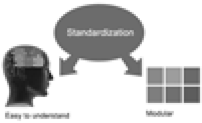
Once FARSC title is accorded, the Fellow is authorized to organize a symposium/seminar/conference on behalf of Global Journal Incorporation (USA).The Fellow can also participate in conference/seminar/symposium organized by another institution as representative of Global Journal. In both the cases, it is mandatory for him to discuss with us and obtain our consent.

You may join as member of the Editorial Board of Global Journals Incorporation (USA) after successful completion of three years as Fellow and as Peer Reviewer. In addition, it is also desirable that you should organize seminar/symposium/conference at least once.

We shall provide you intimation regarding launching of e-version of journal of your stream time to time. This may be utilized in your library for the enrichment of knowledge of your students as well as it can also be helpful for the concerned faculty members.

The FARSC can go through standards of OARS. You can also play vital role if you have any suggestions so that proper amendment can take place to improve the same for the benefit of entire research community.

As FARSC, you will be given a renowned, secure and free professional email address with 100 GB of space e.g. johnhall@globaljournals.org. This will include Webmail, Spam Assassin, Email Forwarders, Auto-Responders, Email Delivery Route tracing, etc.



The FARSC will be eligible for a free application of standardization of their researches. Standardization of research will be subject to acceptability within stipulated norms as the next step after publishing in a journal. We shall depute a team of specialized research professionals who will render their services for elevating your researches to next higher level, which is worldwide open standardization.

The FARSC member can apply for grading and certification of standards of their educational and Institutional Degrees to Open Association of Research, Society U.S.A. Once you are designated as FARSC, you may send us a scanned copy of all of your credentials. OARS will verify, grade and certify them. This will be based on your academic records, quality of research papers published by you, and some more criteria. After certification of all your credentials by OARS, they will be published on your Fellow Profile link on website https://associationofresearch.org which will be helpful to upgrade the dignity.

The FARSC members can avail the benefits of free research podcasting in Global Research Radio with their research documents. After publishing the work, (including published elsewhere worldwide with proper authorization) you can upload your research paper with your recorded voice or you can utilize chargeable services of our professional RJs to record your paper in their voice on request.

The FARSC member also entitled to get the benefits of free research podcasting of their research documents through video clips. We can also streamline your conference videos and display your slides/ online slides and online research video clips at reasonable charges, on request.

The FARSC is eligible to earn from sales proceeds of his/her researches/reference/review Books or literature, while publishing with Global Journals. The FARSC can decide whether he/she would like to publish his/her research in a closed manner. In this case, whenever readers purchase that individual research paper for reading, maximum 60% of its profit earned as royalty by Global Journals, will be credited to his/her bank account. The entire entitled amount will be credited to his/her bank account exceeding limit of minimum fixed balance. There is no minimum time limit for collection. The FARSC member can decide its price and we can help in making the right decision.

The FARSC member is eligible to join as a paid peer reviewer at Global Journals Incorporation (USA) and can get remuneration of 15% of author fees, taken from the author of a respective paper. After reviewing 5 or more papers you can request to transfer the amount to your bank account.

MEMBER OF ASSOCIATION OF RESEARCH SOCIETY IN COMPUTING (MARSC)

The ' MARSC ' title is accorded to a selected professional after the approval of the Editor-in-Chief / Editorial Board Members/Dean.
The “MARSC” is a dignified ornament which is accorded to a person’s name viz. Dr. John E. Hall, Ph.D., MARSC or William Walldroff, M.S., MARSC.

MARSC accrediting is an honor. It authenticates your research activities. After becoming MARSC, you can add 'MARSC' title with your name as you use this recognition as additional suffix to your status. This will definitely enhance and add more value and repute to your name. You may use it on your professional Counseling Materials such as CV, Resume, Visiting Card and Name Plate etc.

The following benefitscan be availed by you only for next three years from the date of certification.

MARSC designated members are entitled to avail a 25% discount while publishing their research papers (of a single author) in Global Journals Inc., if the same is accepted by our Editorial Board and Peer Reviewers. If you are a main author or co-author of a group of authors, you will get discount of 10%.

As MARSC, you will be given a renowned, secure and free professional email address with 30 GB of space e.g. johnhall@globaljournals.org. This will include Webmail, Spam Assassin, Email Forwarders, Auto-Responders, Email Delivery Route tracing, etc.

We shall provide you intimation regarding launching of e-version of journal of your stream time to time. This may be utilized in your library for the enrichment of knowledge of your students as well as it can also be helpful for the concerned faculty members.

The MARSC member can apply for approval, grading and certification of standards of their educational and Institutional Degrees to Open Association of Research, Society U.S.A.

Once you are designated as MARSC, you may send us a scanned copy of all of your credentials. OARS will verify, grade and certify them. This will be based on your academic records, quality of research papers published by you, and some more criteria.

It is mandatory to read all terms and conditions carefully.

Auxiliary Memberships

Institutional Fellow of Open Association of Research Society (USA)-OARS (USA)

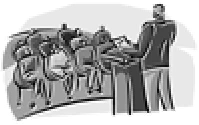
Global Journals Incorporation (USA) is accredited by Open Association of Research Society, U.S.A (OARS) and in turn, affiliates research institutions as "Institutional Fellow of Open Association of Research Society" (IFOARS).

The "FARSC" is a dignified title which is accorded to a person's name viz. Dr. John E. Hall, Ph.D., FARSC or William Walldroff, M.S., FARSC.

The IFOARS institution is entitled to form a Board comprised of one Chairperson and three to five board members preferably from different streams. The Board will be recognized as "Institutional Board of Open Association of Research Society"-(IBOARS).

The Institute will be entitled to following benefits:

The IBOARS can initially review research papers of their institute and recommend them to publish with respective journal of Global Journals. It can also review the papers of other institutions after obtaining our consent. The second review will be done by peer reviewer of Global Journals Incorporation (USA)
The Board is at liberty to appoint a peer reviewer with the approval of chairperson after consulting us.
The author fees of such paper may be waived off up to 40%.



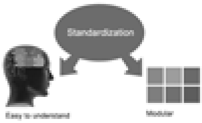
The Global Journals Incorporation (USA) at its discretion can also refer double blind peer reviewed paper at their end to the board for the verification and to get recommendation for final stage of acceptance of publication.

The IBOARS can organize symposium/seminar/conference in their country on behalf of Global Journals Incorporation (USA)-OARS (USA). The terms and conditions can be discussed separately.

The Board can also play vital role by exploring and giving valuable suggestions regarding the Standards of "Open Association of Research Society, U.S.A (OARS)" so that proper amendment can take place for the benefit of entire research community. We shall provide details of particular standard only on receipt of request from the Board.

The board members can also join us as Individual Fellow with 40% discount on total fees applicable to Individual Fellow. They will be entitled to avail all the benefits as declared. Please visit Individual Fellow-sub menu of GlobalJournals.org to have more relevant details.

We shall provide you intimation regarding launching of e-version of journal of your stream time to time. This may be utilized in your library for the enrichment of knowledge of your students as well as it can also be helpful for the concerned faculty members.



After nomination of your institution as “Institutional Fellow” and constantly functioning successfully for one year, we can consider giving recognition to your institute to function as Regional/Zonal office on our behalf.
The board can also take up the additional allied activities for betterment after our consultation.

The following entitlements are applicable to individual Fellows:

Open Association of Research Society, U.S.A (OARS) By-laws states that an individual Fellow may use the designations as applicable, or the corresponding initials. The Credentials of individual Fellow and Associate designations signify that the individual has gained knowledge of the fundamental concepts. One is magnanimous and proficient in an expertise course covering the professional code of conduct, and follows recognized standards of practice.

Open Association of Research Society (US)/ Global Journals Incorporation (USA), as described in Corporate Statements, are educational, research publishing and professional membership organizations. Achieving our individual Fellow or Associate status is based mainly on meeting stated educational research requirements.

Disbursement of 40% Royalty earned through Global Journals : Researcher = 50%, Peer Reviewer = 37.50%, Institution = 12.50% E.g. Out of 40%, the 20% benefit should be passed on to researcher, 15 % benefit towards remuneration should be given to a reviewer and remaining 5% is to be retained by the institution.

We shall provide print version of 12 issues of any three journals [as per your requirement] out of our 38 journals worth $ 2376 USD.

Other:

The individual Fellow and Associate designations accredited by Open Association of Research Society (US) credentials signify guarantees following achievements:

- The professional accredited with Fellow honor, is entitled to various benefits viz. name, fame, honor, regular flow of income, secured bright future, social status etc.

- In addition to above, if one is single author, then entitled to 40% discount on publishing research paper and can get 10%discount if one is co-author or main author among group of authors.
- The Fellow can organize symposium/seminar/conference on behalf of Global Journals Incorporation (USA) and he/she can also attend the same organized by other institutes on behalf of Global Journals.
- The Fellow can become member of Editorial Board Member after completing 3yrs.
- The Fellow can earn 60% of sales proceeds from the sale of reference/review books/literature/publishing of research paper.
- Fellow can also join as paid peer reviewer and earn 15% remuneration of author charges and can also get an opportunity to join as member of the Editorial Board of Global Journals Incorporation (USA)
- • This individual has learned the basic methods of applying those concepts and techniques to common challenging situations. This individual has further demonstrated an in–depth understanding of the application of suitable techniques to a particular area of research practice.

Note :

"

- In future, if the board feels the necessity to change any board member, the same can be done with the consent of the chairperson along with anyone board member without our approval.
- In case, the chairperson needs to be replaced then consent of 2/3rd board members are required and they are also required to jointly pass the resolution copy of which should be sent to us. In such case, it will be compulsory to obtain our approval before replacement.
- In case of "Difference of Opinion [if any]" among the Board members, our decision will be final and binding to everyone.

"

Process of submission of Research Paper

The Area or field of specialization may or may not be of any category as mentioned in 'Scope of Journal' menu of the GlobalJournals.org website. There are 37 Research Journal categorized with Six parental Journals GJCST, GJMR, GJRE, GJMBR, GJSFR, GJHSS. For Authors should prefer the mentioned categories. There are three widely used systems UDC, DDC and LCC. The details are available as 'Knowledge Abstract' at Home page. The major advantage of this coding is that, the research work will be exposed to and shared with all over the world as we are being abstracted and indexed worldwide.

The paper should be in proper format. The format can be downloaded from first page of 'Author Guideline' Menu. The Author is expected to follow the general rules as mentioned in this menu. The paper should be written in MS-Word Format (*.DOC,*.DOCX).

The Author can submit the paper either online or offline. The authors should prefer online submission.Online Submission: There are three ways to submit your paper:

(A) (I) First, register yourself using top right corner of Home page then Login. If you are already registered, then login using your username and password.

(II) Choose corresponding Journal.

(III) Click 'Submit Manuscript'. Fill required information and Upload the paper.

(B) If you are using Internet Explorer, then Direct Submission through Homepage is also available.

(C) If these two are not convenient, and then email the paper directly to dean@globaljournals.org.

Offline Submission: Author can send the typed form of paper by Post. However, online submission should be preferred.

Preferred Author Guidelines

MANUSCRIPT STYLE INSTRUCTION **(Must be strictly followed)**

Page Size: 8.27" X 11'"

- Left Margin: 0.65
- Right Margin: 0.65
- Top Margin: 0.75
- Bottom Margin: 0.75
- Font type of all text should be Swis 721 Lt BT.
- Paper Title should be of Font Size 24 with one Column section.
- Author Name in Font Size of 11 with one column as of Title.
- Abstract Font size of 9 Bold, "Abstract" word in Italic Bold.
- Main Text: Font size 10 with justified two columns section
- Two Column with Equal Column with of 3.38 and Gaping of .2
- First Character must be three lines Drop capped.
- Paragraph before Spacing of 1 pt and After of 0 pt.
- Line Spacing of 1 pt
- Large Images must be in One Column
- Numbering of First Main Headings (Heading 1) must be in Roman Letters, Capital Letter, and Font Size of 10.
- Numbering of Second Main Headings (Heading 2) must be in Alphabets, Italic, and Font Size of 10.

You can use your own standard format also.
Author Guidelines:

1. General,

2. Ethical Guidelines,

3. Submission of Manuscripts,

4. Manuscript's Category,

5. Structure and Format of Manuscript,

6. After Acceptance.

1. GENERAL

Before submitting your research paper, one is advised to go through the details as mentioned in following heads. It will be beneficial, while peer reviewer justify your paper for publication.

Scope

The Global Journals Inc. (US) welcome the submission of original paper, review paper, survey article relevant to the all the streams of Philosophy and knowledge. The Global Journals Inc. (US) is parental platform for Global Journal of Computer Science and Technology, Researches in Engineering, Medical Research, Science Frontier Research, Human Social Science, Management, and Business organization. The choice of specific field can be done otherwise as following in Abstracting and Indexing Page on this Website. As the all Global

Journals Inc. (US) are being abstracted and indexed (in process) by most of the reputed organizations. Topics of only narrow interest will not be accepted unless they have wider potential or consequences.

2. ETHICAL GUIDELINES

Authors should follow the ethical guidelines as mentioned below for publication of research paper and research activities.

Papers are accepted on strict understanding that the material in whole or in part has not been, nor is being, considered for publication elsewhere. If the paper once accepted by Global Journals Inc. (US) and Editorial Board, will become the copyright of the Global Journals Inc. (US).

Authorship: The authors and coauthors should have active contribution to conception design, analysis and interpretation of findings. They should critically review the contents and drafting of the paper. All should approve the final version of the paper before submission

The Global Journals Inc. (US) follows the definition of authorship set up by the Global Academy of Research and Development. According to the Global Academy of R&D authorship, criteria must be based on:

1) Substantial contributions to conception and acquisition of data, analysis and interpretation of the findings.

2) Drafting the paper and revising it critically regarding important academic content.

3) Final approval of the version of the paper to be published.

All authors should have been credited according to their appropriate contribution in research activity and preparing paper. Contributors who do not match the criteria as authors may be mentioned under Acknowledgement.

Acknowledgements: Contributors to the research other than authors credited should be mentioned under acknowledgement. The specifications of the source of funding for the research if appropriate can be included. Suppliers of resources may be mentioned along with address.

Appeal of Decision: The Editorial Board’s decision on publication of the paper is final and cannot be appealed elsewhere.

Permissions: It is the author's responsibility to have prior permission if all or parts of earlier published illustrations are used in this paper.

Please mention proper reference and appropriate acknowledgements wherever expected.

If all or parts of previously published illustrations are used, permission must be taken from the copyright holder concerned. It is the author's responsibility to take these in writing.

Approval for reproduction/modification of any information (including figures and tables) published elsewhere must be obtained by the authors/copyright holders before submission of the manuscript. Contributors (Authors) are responsible for any copyright fee involved.

3. SUBMISSION OF MANUSCRIPTS

Manuscripts should be uploaded via this online submission page. The online submission is most efficient method for submission of papers, as it enables rapid distribution of manuscripts and consequently speeds up the review procedure. It also enables authors to know the status of their own manuscripts by emailing us. Complete instructions for submitting a paper is available below.

Manuscript submission is a systematic procedure and little preparation is required beyond having all parts of your manuscript in a given format and a computer with an Internet connection and a Web browser. Full help and instructions are provided on-screen. As an author, you will be prompted for login and manuscript details as Field of Paper and then to upload your manuscript file(s) according to the instructions.

To avoid postal delays, all transaction is preferred by e-mail. A finished manuscript submission is confirmed by e-mail immediately and your paper enters the editorial process with no postal delays. When a conclusion is made about the publication of your paper by our Editorial Board, revisions can be submitted online with the same procedure, with an occasion to view and respond to all comments.

Complete support for both authors and co-author is provided.

4. MANUSCRIPT'S CATEGORY

Based on potential and nature, the manuscript can be categorized under the following heads:

Original research paper: Such papers are reports of high-level significant original research work.

Review papers: These are concise, significant but helpful and decisive topics for young researchers.

Research articles: These are handled with small investigation and applications.

Research letters: The letters are small and concise comments on previously published matters.

5. STRUCTURE AND FORMAT OF MANUSCRIPT

The recommended size of original research paper is less than seven thousand words, review papers fewer than seven thousands words also.Preparation of research paper or how to write research paper, are major hurdle, while writing manuscript. The research articles and research letters should be fewer than three thousand words, the structure original research paper; sometime review paper should be as follows:

Papers: These are reports of significant research (typically less than 7000 words equivalent, including tables, figures, references), and comprise:

(a)Title should be relevant and commensurate with the theme of the paper.

(b) A brief Summary, "Abstract" (less than 150 words) containing the major results and conclusions.

(c) Up to ten keywords, that precisely identifies the paper's subject, purpose, and focus.

(d) An Introduction, giving necessary background excluding subheadings; objectives must be clearly declared.

(e) Resources and techniques with sufficient complete experimental details (wherever possible by reference) to permit repetition; sources of information must be given and numerical methods must be specified by reference, unless non-standard.

(f) Results should be presented concisely, by well-designed tables and/or figures; the same data may not be used in both; suitable statistical data should be given. All data must be obtained with attention to numerical detail in the planning stage. As reproduced design has been recognized to be important to experiments for a considerable time, the Editor has decided that any paper that appears not to have adequate numerical treatments of the data will be returned un-refereed;

(g) Discussion should cover the implications and consequences, not just recapitulating the results; conclusions should be summarizing.

(h) Brief Acknowledgements.

(i) References in the proper form.

Authors should very cautiously consider the preparation of papers to ensure that they communicate efficiently. Papers are much more likely to be accepted, if they are cautiously designed and laid out, contain few or no errors, are summarizing, and be conventional to the approach and instructions. They will in addition, be published with much less delays than those that require much technical and editorial correction.

The Editorial Board reserves the right to make literary corrections and to make suggestions to improve briefness.

It is vital, that authors take care in submitting a manuscript that is written in simple language and adheres to published guidelines.

Format

Language: The language of publication is UK English. Authors, for whom English is a second language, must have their manuscript efficiently edited by an English-speaking person before submission to make sure that, the English is of high excellence. It is preferable, that manuscripts should be professionally edited.

Standard Usage, Abbreviations, and Units: Spelling and hyphenation should be conventional to The Concise Oxford English Dictionary. Statistics and measurements should at all times be given in figures, e.g. 16 min, except for when the number begins a sentence. When the number does not refer to a unit of measurement it should be spelt in full unless, it is 160 or greater.

Abbreviations supposed to be used carefully. The abbreviated name or expression is supposed to be cited in full at first usage, followed by the conventional abbreviation in parentheses.

Metric SI units are supposed to generally be used excluding where they conflict with current practice or are confusing. For illustration, 1.4 l rather than 1.4 × 10-3 m3, or 4 mm somewhat than 4 × 10-3 m. Chemical formula and solutions must identify the form used, e.g. anhydrous or hydrated, and the concentration must be in clearly defined units. Common species names should be followed by underlines at the first mention. For following use the generic name should be constricted to a single letter, if it is clear.

Structure

All manuscripts submitted to Global Journals Inc. (US), ought to include:

Title: The title page must carry an instructive title that reflects the content, a running title (less than 45 characters together with spaces), names of the authors and co-authors, and the place(s) wherever the work was carried out. The full postal address in addition with the e-mail address of related author must be given. Up to eleven keywords or very brief phrases have to be given to help data retrieval, mining and indexing.

Abstract, used in Original Papers and Reviews:

Optimizing Abstract for Search Engines

Many researchers searching for information online will use search engines such as Google, Yahoo or similar. By optimizing your paper for search engines, you will amplify the chance of someone finding it. This in turn will make it more likely to be viewed and/or cited in a further work. Global Journals Inc. (US) have compiled these guidelines to facilitate you to maximize the web-friendliness of the most public part of your paper.

Key Words

A major linchpin in research work for the writing research paper is the keyword search, which one will employ to find both library and Internet resources.

One must be persistent and creative in using keywords. An effective keyword search requires a strategy and planning a list of possible keywords and phrases to try.

Search engines for most searches, use Boolean searching, which is somewhat different from Internet searches. The Boolean search uses "operators," words (and, or, not, and near) that enable you to expand or narrow your affords. Tips for research paper while preparing research paper are very helpful guideline of research paper.

Choice of key words is first tool of tips to write research paper. Research paper writing is an art.A few tips for deciding as strategically as possible about keyword search:

- One should start brainstorming lists of possible keywords before even begin searching. Think about the most important concepts related to research work. Ask, "What words would a source have to include to be truly valuable in research paper?" Then consider synonyms for the important words.
- It may take the discovery of only one relevant paper to let steer in the right keyword direction because in most databases, the keywords under which a research paper is abstracted are listed with the paper.
- One should avoid outdated words.

Keywords are the key that opens a door to research work sources. Keyword searching is an art in which researcher's skills are bound to improve with experience and time.

Numerical Methods: Numerical methods used should be clear and, where appropriate, supported by references.

Acknowledgements: Please make these as concise as possible.

References

References follow the Harvard scheme of referencing. References in the text should cite the authors' names followed by the time of their publication, unless there are three or more authors when simply the first author's name is quoted followed by et al. unpublished work has to only be cited where necessary, and only in the text. Copies of references in press in other journals have to be supplied with submitted typescripts. It is necessary that all citations and references be carefully checked before submission, as mistakes or omissions will cause delays.

References to information on the World Wide Web can be given, but only if the information is available without charge to readers on an official site. Wikipedia and Similar websites are not allowed where anyone can change the information. Authors will be asked to make available electronic copies of the cited information for inclusion on the Global Journals Inc. (US) homepage at the judgment of the Editorial Board.

The Editorial Board and Global Journals Inc. (US) recommend that, citation of online-published papers and other material should be done via a DOI (digital object identifier). If an author cites anything, which does not have a DOI, they run the risk of the cited material not being noticeable.

The Editorial Board and Global Journals Inc. (US) recommend the use of a tool such as Reference Manager for reference management and formatting.

Tables, Figures and Figure Legends

Tables: Tables should be few in number, cautiously designed, uncrowned, and include only essential data. Each must have an Arabic number, e.g. Table 4, a self-explanatory caption and be on a separate sheet. Vertical lines should not be used.

Figures: Figures are supposed to be submitted as separate files. Always take in a citation in the text for each figure using Arabic numbers, e.g. Fig. 4. Artwork must be submitted online in electronic form by e-mailing them.

Preparation of Electronic Figures for Publication

Even though low quality images are sufficient for review purposes, print publication requires high quality images to prevent the final product being blurred or fuzzy. Submit (or e-mail) EPS (line art) or TIFF (halftone/photographs) files only. MS PowerPoint and Word Graphics are unsuitable for printed pictures. Do not use pixel-oriented software. Scans (TIFF only) should have a resolution of at least 350 dpi (halftone) or 700 to 1100 dpi (line drawings) in relation to the imitation size. Please give the data for figures in black and white or submit a Color Work Agreement Form. EPS files must be saved with fonts embedded (and with a TIFF preview, if possible).

For scanned images, the scanning resolution (at final image size) ought to be as follows to ensure good reproduction: line art: >650 dpi; halftones (including gel photographs) : >350 dpi; figures containing both halftone and line images: >650 dpi.

Color Charges: It is the rule of the Global Journals Inc. (US) for authors to pay the full cost for the reproduction of their color artwork. Hence, please note that, if there is color artwork in your manuscript when it is accepted for publication, we would require you to complete and return a color work agreement form before your paper can be published.

Figure Legends: Self-explanatory legends of all figures should be incorporated separately under the heading 'Legends to Figures'. In the full-text online edition of the journal, figure legends may possibly be truncated in abbreviated links to the full screen version. Therefore, the first 100 characters of any legend should notify the reader, about the key aspects of the figure.

6. AFTER ACCEPTANCE

Upon approval of a paper for publication, the manuscript will be forwarded to the dean, who is responsible for the publication of the Global Journals Inc. (US).

6.1 Proof Corrections

The corresponding author will receive an e-mail alert containing a link to a website or will be attached. A working e-mail address must therefore be provided for the related author.

Acrobat Reader will be required in order to read this file. This software can be downloaded

(Free of charge) from the following website:

www.adobe.com/products/acrobat/readstep2.html. This will facilitate the file to be opened, read on screen, and printed out in order for any corrections to be added. Further instructions will be sent with the proof.

Proofs must be returned to the dean at dean@globaljournals.org within three days of receipt.

As changes to proofs are costly, we inquire that you only correct typesetting errors. All illustrations are retained by the publisher. Please note that the authors are responsible for all statements made in their work, including changes made by the copy editor.

6.2 Early View of Global Journals Inc. (US) (Publication Prior to Print)

The Global Journals Inc. (US) are enclosed by our publishing's Early View service. Early View articles are complete full-text articles sent in advance of their publication. Early View articles are absolute and final. They have been completely reviewed, revised and edited for publication, and the authors' final corrections have been incorporated. Because they are in final form, no changes can be made after sending them. The nature of Early View articles means that they do not yet have volume, issue or page numbers, so Early View articles cannot be cited in the conventional way.

6.3 Author Services

Online production tracking is available for your article through Author Services. Author Services enables authors to track their article - once it has been accepted - through the production process to publication online and in print. Authors can check the status of their articles online and choose to receive automated e-mails at key stages of production. The authors will receive an e-mail with a unique link that enables them to register and have their article automatically added to the system. Please ensure that a complete e-mail address is provided when submitting the manuscript.

6.4 Author Material Archive Policy

Please note that if not specifically requested, publisher will dispose off hardcopy & electronic information submitted, after the two months of publication. If you require the return of any information submitted, please inform the Editorial Board or dean as soon as possible.

6.5 Offprint and Extra Copies

A PDF offprint of the online-published article will be provided free of charge to the related author, and may be distributed according to the Publisher's terms and conditions. Additional paper offprint may be ordered by emailing us at: editor@globaljournals.org .

You must strictly follow above Author Guidelines before submitting your paper or else we will not at all be responsible for any corrections in future in any of the way.

Before start writing a good quality Computer Science Research Paper, let us first understand what is Computer Science Research Paper? So, Computer Science Research Paper is the paper which is written by professionals or scientists who are associated to Computer Science and Information Technology, or doing research study in these areas. If you are novel to this field then you can consult about this field from your supervisor or guide.

TECHNIQUES FOR WRITING A GOOD QUALITY RESEARCH PAPER:

1. Choosing the topic: In most cases, the topic is searched by the interest of author but it can be also suggested by the guides. You can have several topics and then you can judge that in which topic or subject you are finding yourself most comfortable. This can be done by asking several questions to yourself, like Will I be able to carry our search in this area? Will I find all necessary recourses to accomplish the search? Will I be able to find all information in this field area? If the answer of these types of questions will be "Yes" then you can choose that topic. In most of the cases, you may have to conduct the surveys and have to visit several places because this field is related to Computer Science and Information Technology. Also, you may have to do a lot of work to find all rise and falls regarding the various data of that subject. Sometimes, detailed information plays a vital role, instead of short information.

2. Evaluators are human: First thing to remember that evaluators are also human being. They are not only meant for rejecting a paper. They are here to evaluate your paper. So, present your Best.

3. Think Like Evaluators: If you are in a confusion or getting demotivated that your paper will be accepted by evaluators or not, then think and try to evaluate your paper like an Evaluator. Try to understand that what an evaluator wants in your research paper and automatically you will have your answer.

4. Make blueprints of paper: The outline is the plan or framework that will help you to arrange your thoughts. It will make your paper logical. But remember that all points of your outline must be related to the topic you have chosen.

5. Ask your Guides: If you are having any difficulty in your research, then do not hesitate to share your difficulty to your guide (if you have any). They will surely help you out and resolve your doubts. If you can't clarify what exactly you require for your work then ask the supervisor to help you with the alternative. He might also provide you the list of essential readings.

6. Use of computer is recommended: As you are doing research in the field of Computer Science, then this point is quite obvious.

7. Use right software: Always use good quality software packages. If you are not capable to judge good software then you can lose quality of your paper unknowingly. There are various software programs available to help you, which you can get through Internet.

8. Use the Internet for help: An excellent start for your paper can be by using the Google. It is an excellent search engine, where you can have your doubts resolved. You may also read some answers for the frequent question how to write my research paper or find model research paper. From the internet library you can download books. If you have all required books make important reading selecting and analyzing the specified information. Then put together research paper sketch out.

9. Use and get big pictures: Always use encyclopedias, Wikipedia to get pictures so that you can go into the depth.

10. Bookmarks are useful: When you read any book or magazine, you generally use bookmarks, right! It is a good habit, which helps to not to lose your continuity. You should always use bookmarks while searching on Internet also, which will make your search easier.

11. Revise what you wrote: When you write anything, always read it, summarize it and then finalize it.

12. Make all efforts: Make all efforts to mention what you are going to write in your paper. That means always have a good start. Try to mention everything in introduction, that what is the need of a particular research paper. Polish your work by good skill of writing and always give an evaluator, what he wants.

13. Have backups: When you are going to do any important thing like making research paper, you should always have backup copies of it either in your computer or in paper. This will help you to not to lose any of your important.

14. Produce good diagrams of your own: Always try to include good charts or diagrams in your paper to improve quality. Using several and unnecessary diagrams will degrade the quality of your paper by creating "hotchpotch." So always, try to make and include those diagrams, which are made by your own to improve readability and understandability of your paper.

15. Use of direct quotes: When you do research relevant to literature, history or current affairs then use of quotes become essential but if study is relevant to science then use of quotes is not preferable.

16. Use proper verb tense: Use proper verb tenses in your paper. Use past tense, to present those events that happened. Use present tense to indicate events that are going on. Use future tense to indicate future happening events. Use of improper and wrong tenses will confuse the evaluator. Avoid the sentences that are incomplete.

17. Never use online paper: If you are getting any paper on Internet, then never use it as your research paper because it might be possible that evaluator has already seen it or maybe it is outdated version.

18. Pick a good study spot: To do your research studies always try to pick a spot, which is quiet. Every spot is not for studies. Spot that suits you choose it and proceed further.

19. Know what you know: Always try to know, what you know by making objectives. Else, you will be confused and cannot achieve your target.

20. Use good quality grammar: Always use a good quality grammar and use words that will throw positive impact on evaluator. Use of good quality grammar does not mean to use tough words, that for each word the evaluator has to go through dictionary. Do not start sentence with a conjunction. Do not fragment sentences. Eliminate one-word sentences. Ignore passive voice. Do not ever use a big word when a diminutive one would suffice. Verbs have to be in agreement with their subjects. Prepositions are not expressions to finish sentences with. It is incorrect to ever divide an infinitive. Avoid clichés like the disease. Also, always shun irritating alliteration. Use language that is simple and straight forward. put together a neat summary.

21. Arrangement of information: Each section of the main body should start with an opening sentence and there should be a changeover at the end of the section. Give only valid and powerful arguments to your topic. You may also maintain your arguments with records.

22. Never start in last minute: Always start at right time and give enough time to research work. Leaving everything to the last minute will degrade your paper and spoil your work.

23. Multitasking in research is not good: Doing several things at the same time proves bad habit in case of research activity. Research is an area, where everything has a particular time slot. Divide your research work in parts and do particular part in particular time slot.

24. Never copy others' work: Never copy others' work and give it your name because if evaluator has seen it anywhere you will be in trouble.

25. Take proper rest and food: No matter how many hours you spend for your research activity, if you are not taking care of your health then all your efforts will be in vain. For a quality research, study is must, and this can be done by taking proper rest and food.

26. Go for seminars: Attend seminars if the topic is relevant to your research area. Utilize all your resources.

27. Refresh your mind after intervals: Try to give rest to your mind by listening to soft music or by sleeping in intervals. This will also improve your memory.

28. Make colleagues: Always try to make colleagues. No matter how sharper or intelligent you are, if you make colleagues you can have several ideas, which will be helpful for your research.

29. Think technically: Always think technically. If anything happens, then search its reasons, its benefits, and demerits.

30. Think and then print: When you will go to print your paper, notice that tables are not be split, headings are not detached from their descriptions, and page sequence is maintained.

31. Adding unnecessary information: Do not add unnecessary information, like, I have used MS Excel to draw graph. Do not add irrelevant and inappropriate material. These all will create superfluous. Foreign terminology and phrases are not apropos. One should NEVER take a broad view. Analogy in script is like feathers on a snake. Not at all use a large word when a very small one would be sufficient. Use words properly, regardless of how others use them. Remove quotations. Puns are for kids, not grunt readers. Amplification is a billion times of inferior quality than sarcasm.

32. Never oversimplify everything: To add material in your research paper, never go for oversimplification. This will definitely irritate the evaluator. Be more or less specific. Also too, by no means, ever use rhythmic redundancies. Contractions aren't essential and shouldn't be there used. Comparisons are as terrible as clichés. Give up ampersands and abbreviations, and so on. Remove commas, that are, not necessary. Parenthetical words however should be together with this in commas. Understatement is all the time the complete best way to put onward earth-shaking thoughts. Give a detailed literary review.

33. Report concluded results: Use concluded results. From raw data, filter the results and then conclude your studies based on measurements and observations taken. Significant figures and appropriate number of decimal places should be used. Parenthetical remarks are prohibitive. Proofread carefully at final stage. In the end give outline to your arguments. Spot out perspectives of further study of this subject. Justify your conclusion by at the bottom of them with sufficient justifications and examples.

34. After conclusion: Once you have concluded your research, the next most important step is to present your findings. Presentation is extremely important as it is the definite medium though which your research is going to be in print to the rest of the crowd. Care should be taken to categorize your thoughts well and present them in a logical and neat manner. A good quality research paper format is essential because it serves to highlight your research paper and bring to light all necessary aspects in your research.

Informal Guidelines of Research Paper Writing

Key points to remember:

- Submit all work in its final form.
- Write your paper in the form, which is presented in the guidelines using the template.
- Please note the criterion for grading the final paper by peer-reviewers.

Final Points:

A purpose of organizing a research paper is to let people to interpret your effort selectively. The journal requires the following sections, submitted in the order listed, each section to start on a new page.

The introduction will be compiled from reference matter and will reflect the design processes or outline of basis that direct you to make study. As you will carry out the process of study, the method and process section will be constructed as like that. The result segment will show related statistics in nearly sequential order and will direct the reviewers next to the similar intellectual paths throughout the data that you took to carry out your study. The discussion section will provide understanding of the data and projections as to the implication of the results. The use of good quality references all through the paper will give the effort trustworthiness by representing an alertness of prior workings.

Writing a research paper is not an easy job no matter how trouble-free the actual research or concept. Practice, excellent preparation, and controlled record keeping are the only means to make straightforward the progression.

General style:

Specific editorial column necessities for compliance of a manuscript will always take over from directions in these general guidelines.

To make a paper clear

· Adhere to recommended page limits

Mistakes to evade

- Insertion a title at the foot of a page with the subsequent text on the next page
- Separating a table/chart or figure - impound each figure/table to a single page
- Submitting a manuscript with pages out of sequence

In every sections of your document

· Use standard writing style including articles ("a", "the," etc.)

· Keep on paying attention on the research topic of the paper

· Use paragraphs to split each significant point (excluding for the abstract)

· Align the primary line of each section

· Present your points in sound order

· Use present tense to report well accepted

· Use past tense to describe specific results

· Shun familiar wording, don't address the reviewer directly, and don't use slang, slang language, or superlatives

· Shun use of extra pictures - include only those figures essential to presenting results

Title Page:

Choose a revealing title. It should be short. It should not have non-standard acronyms or abbreviations. It should not exceed two printed lines. It should include the name(s) and address (es) of all authors.

Abstract:

The summary should be two hundred words or less. It should briefly and clearly explain the key findings reported in the manuscript--must have precise statistics. It should not have abnormal acronyms or abbreviations. It should be logical in itself. Shun citing references at this point.

An abstract is a brief distinct paragraph summary of finished work or work in development. In a minute or less a reviewer can be taught the foundation behind the study, common approach to the problem, relevant results, and significant conclusions or new questions.

Write your summary when your paper is completed because how can you write the summary of anything which is not yet written? Wealth of terminology is very essential in abstract. Yet, use comprehensive sentences and do not let go readability for briefness. You can maintain it succinct by phrasing sentences so that they provide more than lone rationale. The author can at this moment go straight to shortening the outcome. Sum up the study, with the subsequent elements in any summary. Try to maintain the initial two items to no more than one ruling each.

- Reason of the study - theory, overall issue, purpose
- Fundamental goal
- To the point depiction of the research
- Consequences, including definite statistics - if the consequences are quantitative in nature, account quantitative data; results of any numerical analysis should be reported
- Significant conclusions or questions that track from the research(es)

Approach:

- Single section, and succinct
- As a outline of job done, it is always written in past tense
- A conceptual should situate on its own, and not submit to any other part of the paper such as a form or table
- Center on shortening results - bound background information to a verdict or two, if completely necessary
- What you account in an conceptual must be regular with what you reported in the manuscript
- Exact spelling, clearness of sentences and phrases, and appropriate reporting of quantities (proper units, important statistics) are just as significant in an abstract as they are anywhere else

Introduction:

The **Introduction** should "introduce" the manuscript. The reviewer should be presented with sufficient background information to be capable to comprehend and calculate the purpose of your study without having to submit to other works. The basis for the study should be offered. Give most important references but shun difficult to make a comprehensive appraisal of the topic. In the introduction, describe the problem visibly. If the problem is not acknowledged in a logical, reasonable way, the reviewer will have no attention in your result. Speak in common terms about techniques used to explain the problem, if needed, but do not present any particulars about the protocols here. Following approach can create a valuable beginning:

- Explain the value (significance) of the study
- Shield the model - why did you employ this particular system or method? What is its compensation? You strength remark on its appropriateness from a abstract point of vision as well as point out sensible reasons for using it.
- Present a justification. Status your particular theory (es) or aim(s), and describe the logic that led you to choose them.
- Very for a short time explain the tentative propose and how it skilled the declared objectives.

Approach:

- Use past tense except for when referring to recognized facts. After all, the manuscript will be submitted after the entire job is done.
- Sort out your thoughts; manufacture one key point with every section. If you make the four points listed above, you will need a least of four paragraphs.

- Present surroundings information only as desirable in order hold up a situation. The reviewer does not desire to read the whole thing you know about a topic.
- Shape the theory/purpose specifically - do not take a broad view.
- As always, give awareness to spelling, simplicity and correctness of sentences and phrases.

Procedures (Methods and Materials):

This part is supposed to be the easiest to carve if you have good skills. A sound written Procedures segment allows a capable scientist to replacement your results. Present precise information about your supplies. The suppliers and clarity of reagents can be helpful bits of information. Present methods in sequential order but linked methodologies can be grouped as a segment. Be concise when relating the protocols. Attempt for the least amount of information that would permit another capable scientist to spare your outcome but be cautious that vital information is integrated. The use of subheadings is suggested and ought to be synchronized with the results section. When a technique is used that has been well described in another object, mention the specific item describing a way but draw the basic principle while stating the situation. The purpose is to text all particular resources and broad procedures, so that another person may use some or all of the methods in one more study or referee the scientific value of your work. It is not to be a step by step report of the whole thing you did, nor is a methods section a set of orders.

Materials:

- Explain materials individually only if the study is so complex that it saves liberty this way.
- Embrace particular materials, and any tools or provisions that are not frequently found in laboratories.
- Do not take in frequently found.
- If use of a definite type of tools.
- Materials may be reported in a part section or else they may be recognized along with your measures.

Methods:

- Report the method (not particulars of each process that engaged the same methodology)
- Describe the method entirely
- To be succinct, present methods under headings dedicated to specific dealings or groups of measures
- Simplify - details how procedures were completed not how they were exclusively performed on a particular day.
- If well known procedures were used, account the procedure by name, possibly with reference, and that's all.

Approach:

- It is embarrassed or not possible to use vigorous voice when documenting methods with no using first person, which would focus the reviewer's interest on the researcher rather than the job. As a result when script up the methods most authors use third person passive voice.
- Use standard style in this and in every other part of the paper - avoid familiar lists, and use full sentences.

What to keep away from

- Resources and methods are not a set of information.
- Skip all descriptive information and surroundings - save it for the argument.
- Leave out information that is immaterial to a third party.

Results:

The principle of a results segment is to present and demonstrate your conclusion. Create this part a entirely objective details of the outcome, and save all understanding for the discussion.

The page length of this segment is set by the sum and types of data to be reported. Carry on to be to the point, by means of statistics and tables, if suitable, to present consequences most efficiently.You must obviously differentiate material that would usually be incorporated in a study editorial from any unprocessed data or additional appendix matter that would not be available. In fact, such matter should not be submitted at all except requested by the instructor.

Content

- Sum up your conclusion in text and demonstrate them, if suitable, with figures and tables.
- In manuscript, explain each of your consequences, point the reader to remarks that are most appropriate.
- Present a background, such as by describing the question that was addressed by creation an exacting study.
- Explain results of control experiments and comprise remarks that are not accessible in a prescribed figure or table, if appropriate.
- Examine your data, then prepare the analyzed (transformed) data in the form of a figure (graph), table, or in manuscript form.

What to stay away from

- Do not discuss or infer your outcome, report surroundings information, or try to explain anything.
- Not at all, take in raw data or intermediate calculations in a research manuscript.
- Do not present the similar data more than once.
- Manuscript should complement any figures or tables, not duplicate the identical information.
- Never confuse figures with tables - there is a difference.

Approach

- As forever, use past tense when you submit to your results, and put the whole thing in a reasonable order.
- Put figures and tables, appropriately numbered, in order at the end of the report
- If you desire, you may place your figures and tables properly within the text of your results part.

Figures and tables

- If you put figures and tables at the end of the details, make certain that they are visibly distinguished from any attach appendix materials, such as raw facts
- Despite of position, each figure must be numbered one after the other and complete with subtitle
- In spite of position, each table must be titled, numbered one after the other and complete with heading
- All figure and table must be adequately complete that it could situate on its own, divide from text

Discussion:

The Discussion is expected the trickiest segment to write and describe. A lot of papers submitted for journal are discarded based on problems with the Discussion. There is no head of state for how long a argument should be. Position your understanding of the outcome visibly to lead the reviewer through your conclusions, and then finish the paper with a summing up of the implication of the study. The purpose here is to offer an understanding of your results and hold up for all of your conclusions, using facts from your research and generally accepted information, if suitable. The implication of result should be visibly described. Infer your data in the conversation in suitable depth. This means that when you clarify an observable fact you must explain mechanisms that may account for the observation. If your results vary from your prospect, make clear why that may have happened. If your results agree, then explain the theory that the proof supported. It is never suitable to just state that the data approved with prospect, and let it drop at that.

- Make a decision if each premise is supported, discarded, or if you cannot make a conclusion with assurance. Do not just dismiss a study or part of a study as "uncertain."
- Research papers are not acknowledged if the work is imperfect. Draw what conclusions you can based upon the results that you have, and take care of the study as a finished work
- You may propose future guidelines, such as how the experiment might be personalized to accomplish a new idea.
- Give details all of your remarks as much as possible, focus on mechanisms.
- Make a decision if the tentative design sufficiently addressed the theory, and whether or not it was correctly restricted.
- Try to present substitute explanations if sensible alternatives be present.
- One research will not counter an overall question, so maintain the large picture in mind, where do you go next? The best studies unlock new avenues of study. What questions remain?
- Recommendations for detailed papers will offer supplementary suggestions.

Approach:

- When you refer to information, differentiate data generated by your own studies from available information
- Submit to work done by specific persons (including you) in past tense.
- Submit to generally acknowledged facts and main beliefs in present tense.

The Administration Rules

Please carefully note down following rules and regulation before submitting your Research Paper to Global Journals Inc. (US):

Segment Draft and Final Research Paper: You have to strictly follow the template of research paper. If it is not done your paper may get rejected.

- The **major constraint** is that you must independently make all content, tables, graphs, and facts that are offered in the paper. You must write each part of the paper wholly on your own. The Peer-reviewers need to identify your own perceptive of the concepts in your own terms. NEVER extract straight from any foundation, and never rephrase someone else's analysis.

- Do not give permission to anyone else to "PROOFREAD" your manuscript.

- Methods to avoid Plagiarism is applied by us on every paper, if found guilty, you will be blacklisted by all of our collaborated research groups, your institution will be informed for this and strict legal actions will be taken immediately.)
- To guard yourself and others from possible illegal use please do not permit anyone right to use to your paper and files.

Criterion for Grading a Research Paper (Compilation) by Global Journals Inc. (US)

Please note that following table is only a Grading of "Paper Compilation" and not on "Performed/Stated Research" whose grading solely depends on Individual Assigned Peer Reviewer and Editorial Board Member. These can be available only on request and after decision of Paper. This report will be the property of Global Journals Inc. (US).

Topics	Grades		
	A-B	**C-D**	**E-F**
Abstract	Clear and concise with appropriate content, Correct format. 200 words or below	Unclear summary and no specific data, Incorrect form Above 200 words	No specific data with ambiguous information Above 250 words
Introduction	Containing all background details with clear goal and appropriate details, flow specification, no grammar and spelling mistake, well organized sentence and paragraph, reference cited	Unclear and confusing data, appropriate format, grammar and spelling errors with unorganized matter	Out of place depth and content, hazy format
Methods and Procedures	Clear and to the point with well arranged paragraph, precision and accuracy of facts and figures, well organized subheads	Difficult to comprehend with embarrassed text, too much explanation but completed	Incorrect and unorganized structure with hazy meaning
Result	Well organized, Clear and specific, Correct units with precision, correct data, well structuring of paragraph, no grammar and spelling mistake	Complete and embarrassed text, difficult to comprehend	Irregular format with wrong facts and figures
Discussion	Well organized, meaningful specification, sound conclusion, logical and concise explanation, highly structured paragraph reference cited	Wordy, unclear conclusion, spurious	Conclusion is not cited, unorganized, difficult to comprehend
References	Complete and correct format, well organized	Beside the point, Incomplete	Wrong format and structuring

INDEX

save our planet

Global Journal of Computer Science and Technology

Visit us on the Web at www.GlobalJournals.org | www.ComputerResearch.org
or email us at helpdesk@globaljournals.org

9 70116 58698 2 61427>

ISSN 9754350