SOFTWARE METRICS USING GENETIC ALGORITHM

Reena Sharma[#1] R.S.Chhillar[#2]

Department of computer science & application
M.D. university, Rohtak-124001, Haryana,india

ABSTRACT

Softwares should be robust and reliable. In order to be so there should be a method to measure the program length and volume. The present methods treat all the operators and operands with equal importance, whereas the operator of a module having high priority should be given more importance. The priority of a module can be decided using coupling. The work presented proposes an original method of accomplishing the above task. The work makes use of Genetic Algorithms, as these are one of the most robust heuristic search methods known. Thus, the work presented opens the window of artificial intelligence to software metrics.

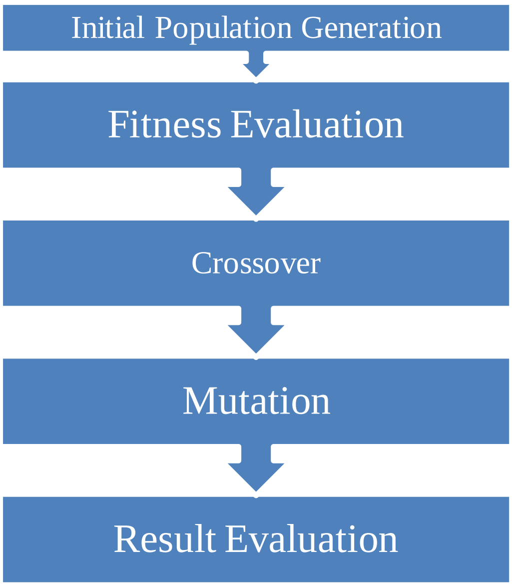
Keywords: Genetic Algorithms, Software Metrics, Program Volume.

INTRODUCTION

It is said that 'what can be measured can be studied'. So, measurement is an important process is any type of engineering. Unlike other engineering disciplines, software engineering is not grounded in the basic quantitative laws of physics. Absolute measurements, such as voltage, mass, velocity or temperature, are uncommon in the software world. Instead, we attempt to derive a set of indirect measures that lead to metrics that provide an indication of the quality of some representation of software. Realizing the importance of software metrics, number of metrics has been defined for software [6]. The use of software metrics is primarily to determine the quality of software. However, they are also used not only to predict the quality of product or process, but also to improve the quality. The goal of the present work is obtaining objective, reproducible and quantifiable measurements, which may have numerous valuable applications. The work is the continuation of our previous work [1]. The work proposes the use of genetic Algorithms in Metrics. Genetic Algorithms are based on the theory of natural selection and the survival of the fittest [2]. The idea is to prioritize the various elements in software and then apply GA to select the fittest whose value. The work intends to introduce the use of Artificial Intelligence and particularly the theory of natural selection to software metrics. The goal is to have well defined metrics even for huge programs. The length and the volume of a program can be measured by the process described in the work. The measurements take into account the importance of a module as per as coupling is concerned. The sections that follow explain the Genetic Process followed by the description of the proposed work and the future scope. Many programs have been analyzed using the techniques. It is definitely not the last word but can be a game changer as per as software metrics are concerned. The work is applicable for large programs with large number of modules as Genetic Algorithms find the fittest chromosome from amongst a large number of chromosomes.

RELATED WORK

Genetic Algorithms are search algorithms based on the theory of the survival of the fittest [2]. Genetic algorithms help to find the right chromosome from amongst large population. John Holland, from the University of Michigan started his work on genetic algorithms at the beginning of the 60s. A first achievement was the publication of Adaptation in Natural and Artificial System in 1975. Holland had two aims, first to improve the understanding of natural adaptation process, second to design artificial systems having properties similar to natural systems. Holland method considers the role of mutation and also utilizes genetic recombination that is crossover to find the optimum solution [5].
The selection is based on the values of a function called fitness function. Fitness function depends on the problem at hand. In our case the fitness function depends on the value assigned to the operators by the second block. The process of Genetic Algorithms has been briefly explained below [2].
Step 1: Firstly, Generate a population of chromosomes. Each chromosome contains cells. The cells can be of any type. In the work proposed the cells are binary numbers. The calls are generated by the pseudo random number generator of the language.
Step 2: Design a fitness function and evaluate individual chromosomes.
Step 3: The next step maps the fitness function to the problem.
GA generally uses the operators: REPRODUCTION operator, Crossover and Mutation operators.
Reproduction is done on the basis of Rowlett Wheel selection .It selects chromosomes from the initial population and enters them into the mating procedure. Crossover Rate determines the probability of producing a new chromosome form the parents. Mutation operator randomly changes its genetic makeup. This operator randomly flips some of the bits in a chromosome. The process has been explained in the following flow graph.



PROPOSED WORK

The work concentrates on developing metrics based the importance of an operator, not just the number. The process starts with the analysis of a program. The program, if procedural, is divided into modules. Modules can be prioritized on the basis of coupling as per a previous work carried out [1]. A new data structure called module table will be developed having the module name, ID and its priority. The operators in the module will be given the same priority as the module in which they appear. The priority of modules is to be decided using Genetic Algorithms (GAs). GAs are heuristic search processes based on the theory of natural selection [2]. The search capabilities of these algorithms have already been established. The priority of a particular module act as a factor, determining the fitness of the module, in the Genetic Process. The process is followed by crossover and mutation. Crossover has been explained in the previous section. The need of mutation arises only if the number of modules is too large. In the analysis the crossover rate is taken as 2% and the mutation rate is taken as 0.5%. Now the operators and the operands are assigned the priority, they will be assigned a number starting from 1. 1 will be assigned to an operator in the module having higher priority, followed by 2 and so on. More is the number, lower is the priority. The reciprocal of that number, therefore, is multiplied by the number of instances of that operator in that module. The total value of N will be

$N = N1*(1/k1) + N2*(1/k2) + N3*(1/K3) + \ldots$ where N1, N2, N3, … are the number of operators and K1, k2, … are the weights assigned to them by the Genetic Priority Assigner.

The value of the Volume will be

$V = V1*(1/k1) + V2*(1/k2) + V3*(1/K3) + \ldots$ where V1, V2, V3, … are the volumes of operators and K1, k2, … are the weights assigned to them by the Genetic Priority Assigner.

The examination and definition of N and V are as follows:

The program vocabulary is given by the number of unique operators plus the number of unique operands [3]

n = n1+n2
n = program vocabulary
n1 = number of unique operators
n2 = number of unique operands

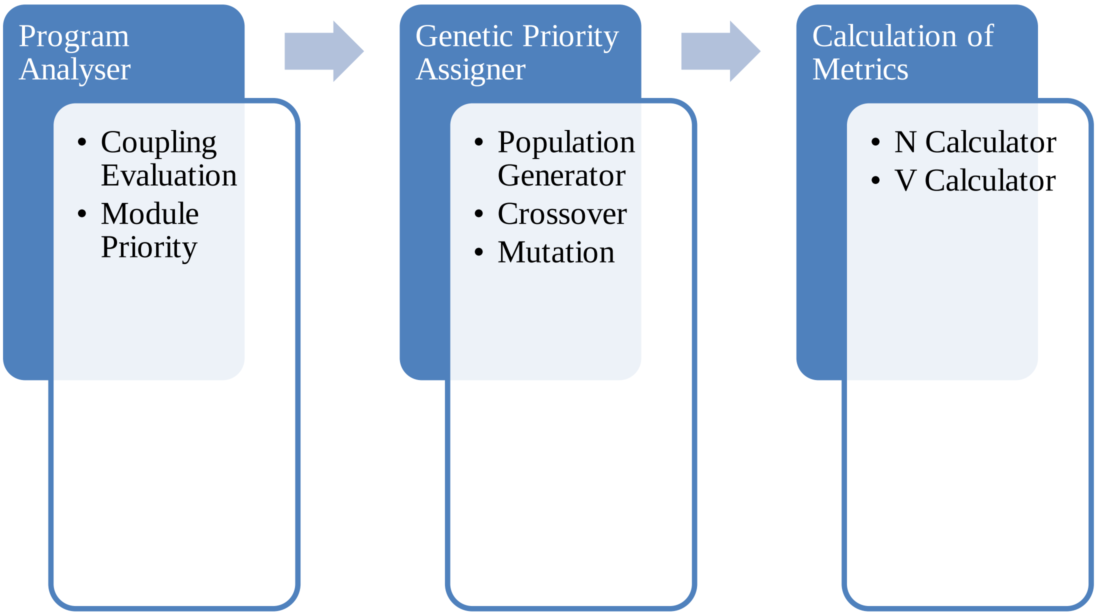
Program Length
The program length is the total usage of all the operators appearing in the implementation plus the total usage of all operands appearing in the implementation [3].
N =N1+N2

N = program length
N1= all operators appearing in the implementation
N2 = all operands appearing in the implementation

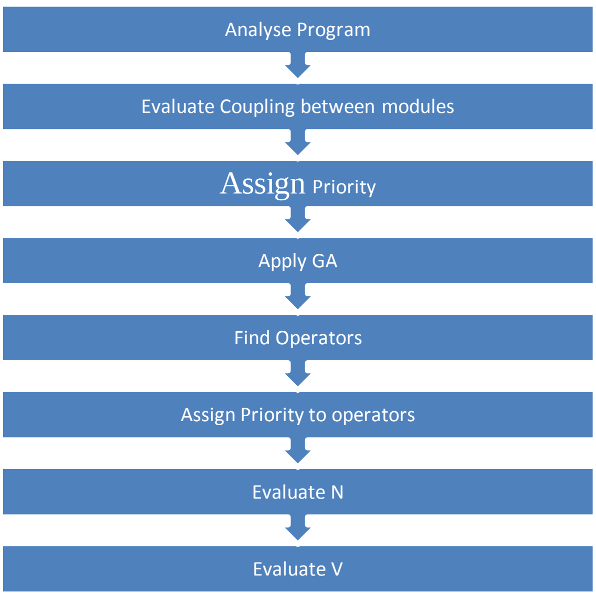
Program Volume
The program volume is defined as the size of the program .This definition is defined by the following equation [3].
V= N log2 n
V = program volume
N = program length

n = program vocabulary
The process has been explained in the flow chart and the block diagrams that follow.

Process



Flow Diagram



RESULTS & CONCLUSIONS

The above work opens a window of the theory of natural selection to the realm of software metrics. A thorough analysis is being done by taking programs of various lengths and number of modules. The effect of the above technique needs to be analyzed and compared with the existing techniques. The Genetic process is sure to find the best results from among vast domain. The robustness of GA is further increased by clubbing it together with the concept of coupling. Moreover, it is our firm believe that the present system of calculating N and V is not that good as it treats each N with same priority. The above process gives a way to treat each N and V with different weights. The process can be altered in case of Object Oriented software as well. The future work will concentrate on the application of GA to the OO systems and their affect. The above work is an honest attempt to reduce the manual work in the evaluation of software metrics and thus is important to make software robust and reliable in future.

REFERENCES

[1]. Reena Sharma, R.S.Chhillar: Novel Approach to Software Metrics. International Journal of Soft Computing and Engineering ,2012.Volume 2, Issue 3,pages 234-236.
[2]. Harsh Bhasin, Surbhi Bhatia: Use of Genetic Algorithms for Finding Roots of Algebraic Equations. International Journal of Computer Science and Information Technology, 2011. Volume 2, Issue 4, pages 693-696.
[3]. Regression Testing Using Coupling and Genetic Algorithms, IJCSIT 3(1) ,Pages : 3255 – 3259, Harsh Bhasin, Manoj.
[4]. Harsh Bhasin, Surbhi Bhatia. Application of Genetic Algorithms in Machine learning, 2011. IJCSIT Volume 2 Issue 5, pages : 2412-2415.
[5]. S. Thrun. Learning to Play the Game of Chess. In G. Tesauro, D. Touretzky, and T. Leen, editors, Advances in Neural Information Processing Systems (NIPS) 7, Cambridge, MA, 1995. MIT Press.
[6]. Empirical Study of Object-Oriented Metrics K.K.Aggarwal, Yogesh Singh, Arvinder Kaur, Ruchika Malhotra JOURNAL OF OBJECT TECHNOLOGY, Online at http://www.jot.fm. Published by ETH Zurich, Chair of Software Engineering ©JOT, 2006.