An Assessment of Halstead Metrics on Health Management Software

K V Krishna*, K V Ganesh
School of Information Technology and Engineering, VIT University, Vellore, India.
*Corresponding author: E-Mail: kv.krishna2013@vit.ac.in

ABSTRACT

This paper mainly describes about the comparison between two-health management software’s of intelligence, effort and time taken to complete the software. Finally, this metric is applied to find the time and effort need for the program to execution and comparison of both the software. This metric will be helpful to choose the appropriate software, which is to save time.

KEY WORDS: Halstead metrics, Health management software, Assembly level code, Effort, Time and cost.

1. INTRODUCTION

Software metrics is one of the software component used to find the value of time, effort of the program need to complete. For example, take COCOMO model, which is used to find the budget, time and effort required to complete the application. Some of the formulas, complexities etc. used as a standard for computation. There are many different kinds of software metrics like Halstead metric, Pearson and Spearman, Control graph, Box plot etc. (Kamsing tso, 2012), Complexity metrics for Health management softwares.

In this paper, we will make use of Halstead metrics to the health management software. Health management software means multiple process can be completed within a short span of time. Not only multitasking but also safe and reliable software. Airplanes, military use this kind of software. The metric must apply to health management equipment code. Programs composed in Assembly, FORTRAN, Pascal, C, and Ada were considered to speak to health management code. Consequently, any of these metric applications could be summed up to apply to health management code. The majority of measurements are chosen in view of their all-inclusive application to code (Kamsing tso, 2012), Complexity metrics for Health management softwares.

Halstead metrics considered the term software science that the physical law measures the code. Software science deals with properties of algorithms whether they are measured directly or indirectly, statically or dynamically. It mainly get the operator and operand as the input from the code and will find the time and effort required for the application to completion (Kamsing tso, 2012), Complexity metrics for Health management softwares.

Halstead metrics applied to the health management software, which is multi-tasking software. The Halstead metric mainly concentrates on the syntax of the application like operators and operands not depends on semantics of the application. The health management software written in assembly level language. Multi-tasking can be done in health management software is difficult to design and implement and comprehend due to real-time constraints.

The Halsted metric is applied to the airline services in which two modules one is defect detection alarm software and another is signal-processing software. Here Some samples of those two modules from various sites are collected in assembly language format and applied the metric to find out the complexity and time for two software. The calculation is done under Numerical analysis part in tabular format. The basic rule in Halstead metric is whatever the language does code given split the operators and operands. Some of the Halstead metric attributes like n1, n2 etc. is calculated. Here we discussed about the comparison of effort and time of two software.

Below given a sample C++ program of how the Halstead metric calculation does, what are all the attributes present, formula for calculation etc. is discussed. This sample code is given to create awareness to the readers about formulas and where it was applied.

Halstead metrics: Halstead metric is one of the software metric which guides the organization to choose the appropriate software in terms of time, effort and complexity of the software.

Intelligence Metric $I = L^V$

Unique number of operators = $n_1$
Unique number of operands = $n_2$
Vocabulary = $n_1 + n_2$
Total number of operators = $N_1$
Total number of operands = $N_2$
Length $L = N_1 + N_2$

$V = N \log_2 n$
$V^* = (2 + n_2) \log_2 (2 + n_2)$
$L = V^*/V$
D = 1 / L
L' = (2 / n1) * (n2 / N2)
I = L' V
E = V / L
T = E / S where S = 18

Sample applications:
Apply the Halstead metric for the following data given:
if(x>0)
cout << “x is positive”
else if (x<0)
cout << “x is –ve”
else
cout <<”x is 0”

Halstead Metrics: For the above data, the following are the attribute values
n1=12
n2=2
Vocabulary = 12+2= 14
Total no of operators N1 = 24
Total no of operands N2 = 4
Length l=N1+N2=28
V = N log n(base 2) =106.6059
V*=0.075
D = 13.325
L’=0.0833
I = 8.883 (Intelligence)
E = 1420.6032 (Effort)
T = E / S =78.93

Table.1. Halstead-defect detection

| Assembly instruction | Operator | Operand |
|----------------------|----------|---------|
| LD A, 0FH            | LD       | A, 0FH  |
| LD (SPSO), A         | LD       | SPSO, A |
| LD A, DONUM          | LD       | A, DONUM|
| LD HL, LSTAT+78H     | LD       | HL, LSTAT + 78H |
| ZEROLP LD HL, 0      | ZEROLP LD| HL, 0   |
| INC HL               | INC      | HL      |
| DEC A                | DEC      | A       |
| JP NZ, ZEROLP        | JP       | NZ, ZEROLP|
| CALL DOSCAN          | CALL     | DOSCAN  |

Table.2. Halstead-file operations

| Assembly instruction | Operator | Operand |
|----------------------|----------|---------|
| MOV EAX, 8           | MOV      | EAX, 8  |
| MOV EBX, FILE_NAME   | MOV      | EBX, FILE_NAME |
| MOV ECX, 0777        | MOV      | ECX, 0777|
| INT 0X80             | INT      | 0X80    |
| MOV [FD_OUT], EAX    | MOV      | [FD_OUT], EAX |
| MOV EDX,LEN          | MOV      | EDX, LEN|
| MOV ECX, MSG         | MOV      | ECX, MSG|
| MOV EBX, [FD_OUT]    | MOV      | EBX, [FD_OUT] |
| MOV EAX, 4           | MOV      | EAX, 4  |
| INT 0X80             | INT      | 0X80    |
| MOV EAX, 6           | MOV      | EAX, 6  |
| MOV EBX, [FD_OUT]    | MOV      | EBX, [FD_OUT] |
| MOV EAX, 4           | MOV      | EAX, 4  |
These two tables describe about the source code which is in assembly language. The program explains about the file operations like reading a file, writing a file, appending data to the file and finally closing a file. Applying Halstead metrics to the program will separate the operands and operators, calculate the following attributes present in the Halstead metrics.

**Numerical analysis:**

| Halstead metric | Application 1 | Application 2 |
|-----------------|---------------|---------------|
| n_1             | 6             | 2             |
| n_2             | 11            | 17            |
| n               | 17            | 19            |
| N_1             | 10            | 30            |
| N_2             | 16            | 55            |
| N               | 26            | 85            |
| V = N log_2 n   | 106.6         | 359.8         |
| V* = (2+n_2) log_2(2+n_2) | 3.7       | 80.4         |
| L = V* / V      | 0.0347        | 0.22          |
| D = 1 / L       | 29.411        | 4.54          |
| L’ = (2/n_1) * (n_2/N_2) | 0.234    | 0.30          |
| I = L’V        | 24.94         | 111.21        |
| E = V / L      | 3072.046      | 1635.45       |
| T = E / S (S = 18) | 170.67   | 90.85         |

**Visualisation:**

*Figure 1. Bar plot for effort*
*Figure 2. Bar plot for time*
*Figure 3. Box plot for time*
Therefore, for this health management software we calculated the intelligence, time and effort to the completion of the application using Halstead metrics. We can make use of some other metrics like COCOMO model for calculating the budget and time for completion of the project. Some of the complexities values needed to be followed as these are under physical law. Pearson and Spearman for correlation coefficient, Kendal’s metric, box plot metric for plotting a graph instead the customers don’t understand the data by displaying in the numerical values etc.

**DISCUSSION**

Health management software which will perform multiple tasks and safe and reliable software used in airlines and military based applications calculated the effort and time would take approximately for the completion of the project is Halstead metrics. Halstead metrics considered the term software science, which the physical law measures the code.

Software science deals with properties of algorithms whether they are measured directly or indirectly, statically or dynamically. It mainly get the operator and operand as the input from the code and will find the time and effort required for the application to completion (Shen, 2014).

**2. CONCLUSION**

Our theory is that health management s programming implemented in Assembly level code has extraordinary multitasking what's more, genuine time qualities, which are not reflected in the unique Halstead measures. In the above sample applications the effort of defect detecting is more when compared to signal processing and the time taken to complete the application 1(defect detection) is more than application 2(signal processing).

This stare into proposed modifications to these unique measurements that are named Multitasking-Real-time augmentations. So this paper will use to find the effort and time to the real-time applications like flights, military purposes that should be reliable and safe. Finally, the complexity of developing the defect detection application is more when compared to the signal processing application (Shen, 2014).

**REFERENCES**

Chandramowliswaran N, Srinivasan S and Chandra Segar T, A Note on Linear based Set Associative Cache address System, International J. on Computer Science and Engg. (IJCSE), 4 (8), 2012, 1383-1386.

Chandramowliswaran N, Srinivasan S and Chandra Segar T, A Novel scheme for Secured Associative Mapping, The International J. of Computer Science and Applications (TIJCSA), 1 (5), 2012, 1-7.

Chandrasegar Thirumalai, Senthilkumar M, Vaishnavi B, Physicians Medicament using Linear Public Key Crypto System, in International conference on Electrical, Electronics, and Optimization Techniques, ICEEOT, IEEE, 2016.

Dhavachelvan P, Chandra Segar T, Satheskumar K, Evaluation of SOA Complexity Metrics Using Weyuker’s Axioms, IEEE International Advance Computing (IACC), 2009, 2325 – 2329.

Fioravanti F, Nesi P, A method and tool for assessing object-oriented projects and metrics management, Journal of Systems and Software, 53 (2), 2000,111-136.

Chandra Segar Thirumalai, Halstead Metric for Intelligence, Effort, Time predictions, 2017.

Kam sing tso, Myron hecht, Kenneth little john, Complexity metrics for Health management softwares, 2012.

Malathy E, Chandra Segar Thirumalai, Review on non-linear set associative cache design, IJPT, 8 (4), 2016, 5320-5330.

Shen V.Y, Conte S.D and Dunsmore H.E, software science revisited, a critical analysis of the theory and its empirical support, IEEE, 9, 1983, 155-165.
Vaishnavi B, Karthikeyan J, Kiran Yarrakula, Chandrasegar Thirumalai, An Assessment Framework for Precipitation Decision Making Using AHP, International Conference on Electronics and Communication Systems (ICECS), IEEE, 2016.

Vinothini S, Chandra Segar Thirumalai, Vijayaragavan R, Analyzing the performance of AFRA with its traditional routing, IRJET, 2 (2), 2015, 373-382.

Vinothini S, Chandra Segar Thirumalai, Vijayaragavan R, Senthil Kumar M, A Cubic based Set Associative Cache encoded mapping, International Research Journal of Engineering and Technology (IRJET), 2 (2), 2015.