Cognitive Complexity: A Model for Distributing Equivalent Programming Problems

Samrat Kumar Dey*[1]; Syed Salauddin Mohammad Tariq; Md. Shariful Islam; Golam Md. Muradul Bashir

[1]Bangladesh Open University, Gazipur-1705, Bangladesh

*Corresponding Author

Contribution

This research article contributes to the area of complexity measurement of programming problems based on cognitive weight. Proposed Complexity Model helps Facilitator to distribute Equivalent Programming Problems among Learners.

Objective

To build a complexity model which will assist the facilitator allocating the programming problems among the learners by preserving the equivalent level of difficulty.

Introduction

Sometimes During the examination, Equal distribution of programming problems among the students may not be possible. Some finds the problem is very modest and others complain that problem is too complex to solve within limited time boundary. So at this point, the necessity for developing a complexity model which is capable of determining the complexity of a given programs and ensures equal distributions.

Methodology

Cognitive complexity is a new dimension for measuring the complexity of software and also a measurement for psychological and cognitive software complexity which is considered as human intelligence artifact. To comprehend a demonstrated program, we have focused on the basic control structures (BCSs) and architecture of the software, which is discussed in Reference [1].

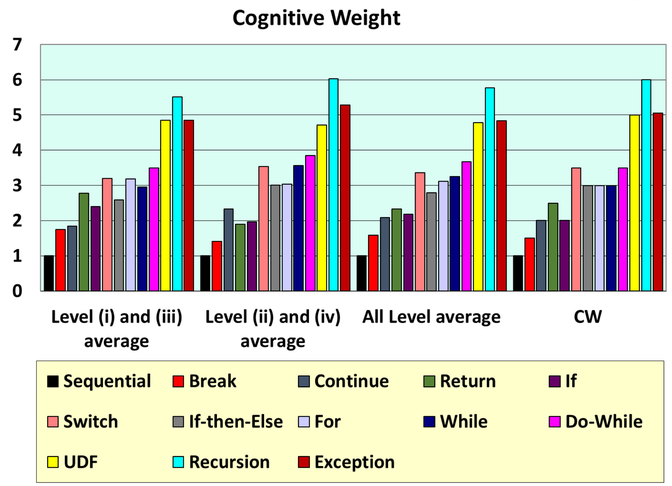

Figure 1. Graphical demonstration of cognitive weight

Reference [2] proposed that for calculating W_{bcs} there exist two different structure

$$Weight_{linear} = \sum_{k=1}^{y} Weight_k \text{ and}$$

$$Weight_{embedded} = \prod_{j=m}^{p} Weight_j$$

Design Architecture

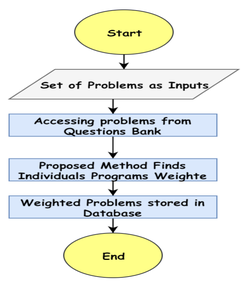

Figure 2. Flowchart of storing weighted problems in Database

Fig.2. represents an initial flow chart indicating how a set of problems is processed through proposed methods of calculating the weight of programming problems. Then weighted problems will have stored in a database for further use in distribution portion.

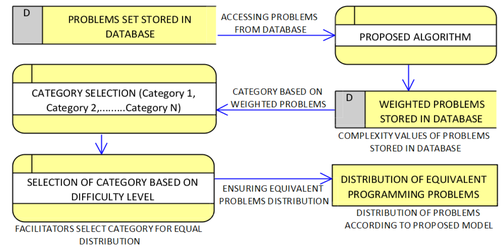

Figure 3. DFD of Proposed Model for Equivalent Problem Distribution

Implementation

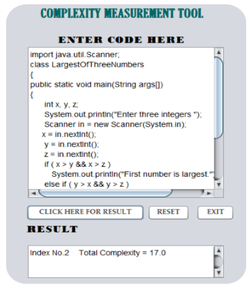

Figure 4. Measuring Complexity value of a java program using developed Tool

Results and Discussions

λ=Lowest weight in a single problem set
α=Highest weight in a single problem set
β=Difference values in α & λ
Θ=facilitator desire category
χ=category count, κ=β/Θ
$\chi_1 = \lambda + \kappa, \chi_2 = \chi_1 + \kappa, \chi_3 = \chi_2 + \kappa, \chi_4 = \chi_3 + \kappa$
.....
Category ω=Category (ω-1) +κ

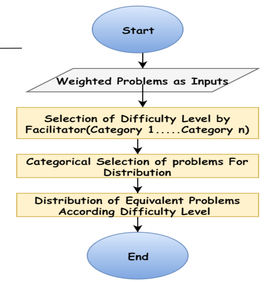

Figure 5. Distributing Programming Problems according difficulty level of category

Conclusion

Some of the momentous features regarding these measurement techniques are a simple calculation of values, consume less time, easily understandable. In future, improvement of this work will produce a boundless dimension

References

1. Wang, Y. August. 2002. On cognitive informatics: Keynote lecture, in Proc. 1st IEEE Int. Conf. Cognitive Informatics (ICCI'02), Calgary, Alta., pp. 34–42
2. Misra, S. 2006.A Complexity Measure Based on Cognitive Weights. *International Journal of Theoretical and Applied Computer Sciences*, Volume 1, Number 1, pp. 1–10.