ANALYSIS OF COGNITIVE COMPLEX LOAD FACTOR ON VISUAL AID BASED COMPUTER PROGRAMMING LANGUAGE LEARNING

MUHAMMED YOUSOOF, MOHD SAPIYAN

1Lecturer, Mathematics and IT Unit, Dhofar University, SULTANATE OF OMAN
2Professor, Faculty of Computer Science and Information Technology, University of Malaya, MALAYSIA

ABSTRACT

Computer applications are dominant and have become part of day to day activities of the human life. These applications are developed by the software developers based on their expertise in domain knowledge and the skills of computer languages. Most of this knowledge of programming has developed from the educational system mainly while learning the computer programming languages at the tertiary educational level such as college education. In general the students are interested and motivated to learn the computer language if provided with effective and user friendly learning tools. The basic fact is that all individuals are different and so their learning capacity differs from one another based on their cognitive ability. This paper addresses the cognitive factors that have the impact during the learning processes and the measurement of cognitive load while the students learn using different computing languages with different tools.

Keywords: Cognitive Complex Factor, Cognitive Load, Computer Programming Language Learning, NASA TLX load

1. INTRODUCTION

Modern Educational system is developing the pedagogy to accommodate the new tools and technology for effective learning process as per the demographic factor, learner and the content. The learning process differs from individual to individual on the basis of age, gender, environment, selection tool and the subject. This paper addresses the learning difficulties of Computer Programming Language (CPL) and the measurement or observation of cognitive load for different visual tools. The next section discusses on the issues on computer CPL learning and its stages of shift in paradigm.

2. REVIEW OF LITERATURE

The literature is replete with recognition that our knowledge-based society demands a different attitude towards learning and its provision, one that spans a lifetime and is integrated into work and other contexts; that our notions of learning must extend beyond the psychological processes of an individual to one that recognizes it is a property of complex systems; and that digital technologies offer a means for realizing complex pedagogies that free formal education from some of the constraints of the past. These insights have influenced recent learning design theory and discourse: constructivist design theories for problem solving (Jonassen, 1999; Reigeluth, 1999; Schwarz, Lin, Brophy, & Bransford, 1999); calls for new directions on technology-based design (Kozma, 2000); design methodologies for acquiring complex skills (van Merriënboer, 1997); and a ‘first principles of instruction’ approach to instructional design (Merrill, 2002). The learning process are balanced with modern learning tools such a computer Aided
multimedia tools and visual aids are selected for effective teaching learning process.

At the same time, much research surrounding multimedia and learning has sought to identify what types of physical media enhance learning (e.g., Clark, 1983; Najjar, 1996; Wu and Martin, 1997). Taken together, it is not surprising to find educational and training systems designed in terms of physical media contained in the system (cf, Jonassen and Mandl, 1990). For example, Stemler (1997) recommends designing educational multimedia systems in terms of elements such as graphics, animations, video and audio. The Demarco system, an art history archive, provides access to digital video without any other contextual information (MacKenzie, 1996). Lastly, consider the common educational Web page which offers its users raw access to simple video or sound clips. In fact, studies that compare learning from different types of media are inconclusive in showing advantages of one physical medium over another. Thus innovative technology has become an integral element of any learning process and CPL is also no exception. Various visualization tools are been in place to teach and learn the CPL effectively. Some of such systems include BlueJ, Ville, Jeliot, Codtwitz Minerva project etc.

3. COMPUTER PROGRAMMING(CPL) LEARNING PROCESS

In the earlier days, programming was mostly procedural and students were first exposed to developing the logic of problem solving. This was supported by using flowcharts and pseudo codes. Mostly the programming languages at the time were BASIC and Pascal. The students after trying to grasp the logic slowly convert them to programs by writing codes. In late 90s the shift of programming was more towards to object oriented programming. When object oriented programming was introduced mostly the learners were taught the concepts. But the problem was the concepts were abstract and not clear to understand for the level of novice. Many programmers tend to face rising challenges when trying to learn the concepts. Later the concept of UML designing was introduced as a course to help understand the object oriented concepts.

Learning theories are the bases which help teachers and educators to understand diverse factors of individual differentiation in: perceiving information; encoding information; transferring information; scanning the representation of the information; and working memory capacity. There are also individual differences in styles of remembering, thinking, and judging, and these individual variations, if not directly part of the personality, are at the very least intimately associated with various non-cognitive dimensions of personality (Kogan N., 1976). Differences in the above factors are brought together to suggest that individuals have different cognitive styles and are different in intelligence, ability, personality, and achievement. It seems that our cognitive style influences our: intellectual abilities; skills; personalities; teaching and learning; and performance. According to (Messick's 1993) definition “cognitive styles are characteristic modes of perceiving, remembering, thinking, problem solving, decision making that are reflective of information processing regularities that develop in congenial ways”

It is not easy to understand the difficulty faced by the learners as human learning process is complex and in our case the programming as a skill is also complex. The novice programmers need to acquire the many skills while learning computer programming language. It takes a time for the learners to become experts to build proper mental model or schema in the memory of the subject. The learners often miss of fail to understand the steps of programming or concepts since most of them approach coding as line by line sequence and fail to visualize in their mind, what happens and how things works in a program. The learners try to imitate their tuptprs in writing the program and they can write similar programs but often find it hard to implement same technique of program coding in another scenario.

In general, programming is taught to learners by using the emphasis of syntax and most of the time learners don’t find it difficult to understand and illustrate the syntax of the programming concept.
when writing the program. But they find difficult is that they could not understand when and how to implement this syntax when executing the program to make it as building blocks for the programs. This could be attributed due to the fact that basis of program learning is problem solving and method of execution such as sequence, control and object based approach. This basic problem occurs based on the mapping process of the learner memory and it mapping of the execution of the coding.

According to (Osborne M., (2004)) Many learners often fail to develop mechanisms and explanations in the process of learning. Programming involves gaining knowledge and using appropriate skills to build the objects. Mechanisms turn the static program into dynamic representation showing causal relationships between statements and show how the program executes. The programming course is not only concerned about developing the skills of writing program codes but also the application of the program coding in various situations of problem solving using computers using procedural and control coding. But most of the time the learners find it hard to do this and there is a block in the transfer of knowledge of application of problem solving skills to write a complete program. In some cases the users are unable to combine the various constructs of programming to build an appropriate program.

The above stated problems could be easily related to the Cognitive Load Theory (Taber K.S., (2003), Terrell S.T., (2002), and Struyven K., Dochy F. and Janssens S. (2002)). Cognitive Load theory justifies the reason why experts outperform novice is due to lack of schemata formation and inability to process a lot of information due to limited capacity of the working memory. There is also a lack of mathematical skills which forms a base to solve many real time problems. Personal factors like aptitude and motivation also play a role. But this is not a unique difficulty for programming, as it is commonly experienced for any learning. Learners often lack the skills necessary in the beginning stages of learning.

Programming has a common problem like mathematics. Mathematics is often seen as difficult subject by many learners. Both mathematics and programming involve lot of abstraction and not very clear from learner’s perspective. It needs to develop analytical skills to solve problems. The development of the skills is hindered due to the complexity and various terms used. But once the concept is understood it remains intact in the Long term memory. This can be applied to any new problems or scenarios.

Cognitive load plays a major role in the learning process. The learner’s emotional, intelligent processes are interrelated in the learning process. In this paper, using the NASA TLX scale rating method cognitive work load is calculated for each process in different programming language with heterogeneous group.

4. METHODOLOGY

This research analysis process is based on the principles of scientific research process. The learners are selected from the existing students population those who are doing under graduation. Out of the selected population some are new to the computer programming language; some studied the computer programming language from Intermediate or Pre University levels. These selected groups of students’ basic language skill tested to classify into groups based on their level of mastery of the English language. The learners are divided into three groups one group learnt using traditional methods and the other two groups used computer visual tools namely Ville and Teaching Machine in for learning process. Their performance and their cognitive load are measured and analyzed in the later sections of this paper. The following diagram illustrates the process of Cognitive load measurement by observing the learning performance using different test instruments at various levels and co relates with the Cognitive Load observation.
5. EXPERIMENT DESIGN

The cognitive learning factor work load is observed with 40 samples. The learning contents varies from group to group. The first group learnt C programming, the second group learnt C++ as they are familiar with C and third group learnt C++ language. Out of 40 students 16 are boys and 24 are girls. Thus a balanced approach for gender is given in our experiments. Each of the sample is made to learn 6 conceptual modules such as variable, condition, looping, array, functions and file handlings irrespective of the programming language which they are learning. The demographic distribution of the learners is given below 28% from Urban, 35% from the Semi Urban and 38% from rural schooling system. In the group, 43% of the learners had prior computing knowledge and fundamentals of Computer with basic computing knowledge from Pre University course. 27% of learners had no prior math and computer Science exposure. 30% of the learners are from commence and computer science as a subject in the pre university.

The selected populations are tested with their Basic English language understanding using stand IELTS examination as we found that it will help to determine their level of understanding the concepts and the essentials of computer programming. The language is a vital element in the learning process as it is the medium of the learning process.

6. NASA GRADING SCALE FOR COGNITIVE FACTORS

NASA task load index is a multidimensional rating procedure that provides over all work load based on a weighted average of rating on six subscales which are mental demand, physical demand, temporal demand, Own performance, Effort and Frustration [Hart, S. G., & Staveland, L. E. (1988). Beyers, J.C., Bittner, Jr, A. and Hills, S.G 1989., Wierwille, W.W. & Eggemeier, F.T. (1993), Hitt II, J.M., Kring, J.P., Daskarolis, E., Morris, C., & Mouloua, M. (1999).] This technique (referred to as the “NASA Bipolar Rating Scale”) was quite successful in reducing between-rater variability and provided diagnostic information about the magnitudes of different sources of load from subscale ratings (Hart, Battiste & Lester 1984. Vidulich & Tsang 1985a & b). However, its sensitivity to experimental manipulations, while better than found for other popular techniques and as a global one-dimensional workload rating.

Mental demand deals with mental and perceptual activity was required for learning the concept and implementation CPL. Physical demand focused physical activity was required. Temporal demand deals with time pressure of the students while learning the CPL. Performance shows the accomplishment of goals of the task set by the experimenter. Frustration level specifies the insecure, discouraged, irritated, stressed and annoyed versus of the learning. Effort measures the work to accomplish your level of performance.

7. EXPERIMENT

From the selected learners, three test groups were formed namely the ville, Teaching machine and traditional classroom group and this group of students were taught the following programming languages which is C, C++ and C to C++(students with prior knowledge of C Programming already) Computer program language learners. The follow paragraph gives an overview about the two visualization tools used in
the experiment which are Ville and teaching machine.

VILLE is a language-independent program visualization tool providing an abstract view of programming. It can be used both in lectures and for independent learning. It has a built-in syntax editor with which users can add new languages to the tool or modify the syntax of built-in languages (currently including Java, C++, and a pseudolanguage). The visualizations can be viewed in any of the defined languages. To emphasize the language independency, VILLE has a parallel view displaying a program in two languages simultaneously. It is possible to trace program execution line by line and monitor program outputs and changes in variable values. To make visualization more effective and easily interpretable, there is an automatically generated textual description of each code line, including the role information of variables. VILLE comes with a set of predefined examples, which can be easily extended. In addition, VILLE’s predefined or user-defined examples can be published on the web, allowing students to engage with a learning session at any time and place.

The Teaching Machine is a teaching and learning tool designed to harness the power of computers and the web to improve student's understanding of programming. It is based on, and designed to develop in students, a mental model of computing that we believe represents how experienced programmers see computers. It has been used as an aid to teach programming in the Faculty of Engineering at Memorial since 1999, with very positive results. The tool works by interactively animating C++ and Java programs. Written in Java, it can be run either as a stand-alone program or as an applet. The class room method is implemented with the teacher and text book in a traditional manner. In our experiments each group and individual’s learning ability is evaluated and their performance is measured as discussed before using NASA TLX scale.

8. CALCULATION OF COGNITIVE LOAD

The NASA Task Load Index workload evaluation procedure is a two-part procedure requiring the collection of both weights and ratings from the subject and the manipulation of the collected data to provide weighted subscale ratings and an Overall Workload score. There are fifteen possible pair-wise comparisons of the six scale elements. When Weights is selected the load presents each pair to the subject one pair at a time. The order in which the pairs are presented and the position of the two elements (left or right) are completely randomized and are different. Factor select which element they felt contributed the most to the workload on the specific task. When all fifteen possible pairs have been presented the second part will be continue.

The second requirement is to obtain numerical ratings for each scale element that reflect the magnitude of that factor. The subject responds between 0 to 20. The Weighted workload rating for each element in a task is simply the Weight (tally) for that element a number between zero and five, multiplied by the Magnitude of load, a number between zero and one hundred. Therefore if the subject had totaled 4 for the weight of Temporal Demand and indicated a magnitude of Temporal Demand in a particular task to be 45, the weighted workload due to Temporal Demand for that particular task would be 90. OVERALL workload for a particular task is determined by summing all of the weighted workload ratings for an individual subject for the particular task and dividing by 15. Using the above calculation the different combinational learning and cognitive load are observed and calculated.
Weighted rating determined from the sum of sum of adjusted rating and divided with 15. As per the calculation the learners

9. INTERPERATION AND ANALYSIS

The research carried out with a sample of 40 learners over six month duration in three subjects namely C, C++ and C to C++ course (prior knowledge of C) covering a variety of concepts in programming. The three categories of students have been classified and subjected to experience using the ville, teaching machine tools and traditional class room methods were used. In the first language c learners experienced an average of 57% cognitive work load and secured the performance of 13 points. In the class room initially they faced more difficulties. The cognitive load is less in the minimum level when using teaching machine tool when compared to the ville. But in the average ville tool is more suitable to learn for C language because the maximum cognitive load is 53%.

<table>
<thead>
<tr>
<th>Subject</th>
<th>Method/Tools</th>
<th>Number of Learners</th>
<th>Cognitive Work Load</th>
<th>Learning Performance</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>ville</td>
<td>5</td>
<td>40</td>
<td>70</td>
</tr>
<tr>
<td>C</td>
<td>teaching machine</td>
<td>5</td>
<td>38</td>
<td>75</td>
</tr>
<tr>
<td>C</td>
<td>class room methods</td>
<td>5</td>
<td>42</td>
<td>77</td>
</tr>
<tr>
<td>C++</td>
<td>ville</td>
<td>5</td>
<td>39</td>
<td>69</td>
</tr>
<tr>
<td>C++</td>
<td>teaching machine</td>
<td>5</td>
<td>43</td>
<td>68</td>
</tr>
<tr>
<td>C++</td>
<td>class room methods</td>
<td>4</td>
<td>44</td>
<td>77</td>
</tr>
<tr>
<td>C to C++</td>
<td>ville</td>
<td>4</td>
<td>41</td>
<td>72</td>
</tr>
<tr>
<td>C to C++</td>
<td>teaching machine</td>
<td>4</td>
<td>37</td>
<td>76</td>
</tr>
<tr>
<td>C to C++</td>
<td>class room methods</td>
<td>3</td>
<td>44</td>
<td>69</td>
</tr>
</tbody>
</table>

In the C++ learning process teaching machine tool has less performance compared to the ville tool and the students experienced higher cognitive load. While comparing the equal performance of the C++ learning process though the performance is same in the remaining two methods. Therefore cognitive load of teaching machine is less compared to the remaining two methods. Herewith teaching machine is recommended for the learning of C++ language. While analyzing the learning process of the C to C++ (group), the teaching machine performance is good with less cognitive work load and by observing the cognitive work load the visualization tool ville is recommended for the C learners and Teaching machine is recommended for remaining two groups.

10. CONCLUSION

The paper addressed the issues on the challenges and difficulties of the Computer Programming Language. The learners are motivated to learn according to their personal ability. The learning process involved the cognitive factors of mental demand, physical demand, temporal demand, Own performance, Effort and Frustration. This work observed the learning performance of 40 learners and their cognitive load as per the scaling process of NASA. As a result from the selected three methods teaching machine tool is suitable for the computer programming language. It is evident that there is a difference in the student’s performance based on the type of visualization tools and thus the selection of the visualization tool could be customized by using a framework which is implemented using neural network model. Thus the learning can be made effective as it suits according to individual learners.

REFERENCES:


rating scale: Results of empirical and theoretical research. NASA
[5]. P. A. Hancock & N. Meshkati (Eds.), Human mental workload, 139-183. Amsterdam, The Netherlands: Elsevier.
AUTHOR PROFILES:

Mr. Muhammed Yousoof is working as Lecturer in Math and IT Unit, Dhofar University, Sultanate of Oman. He is doing his PhD degree from the University of Malaya, Malaysia.

Prof. Dr. Mohd Sapiyan is a working a professor in the Faculty of Computer science and Information Technology in University of Malaya, Malaysia. He has supervised several PhD and Master students. He has been the Dean of the Faculty of Computer Science & IT and the Director for Center of IT in University of Malaya. His research interests include AI in education and visualization.