

# Measuring Cognitive Load - A Solution to Ease Learning of Programming

Muhammed Yousoof, Mohd Sapiyan, and Khaja Kamaluddin

**Abstract**—Learning programming is difficult for many learners. Some researches have found that the main difficulty relates to cognitive load. Cognitive overload happens in programming due to the nature of the subject which is intrinsically over-bearing on the working memory. It happens due to the complexity of the subject itself. The problem is made worse by the poor instructional design methodology used in the teaching and learning process. Various efforts have been proposed to reduce the cognitive load, e.g. visualization softwares, part-program method etc. Use of many computer based systems have also been tried to tackle the problem. However, little success has been made to alleviate the problem. More has to be done to overcome this hurdle. This research attempts at understanding how cognitive load can be managed so as to reduce the problem of overloading. We propose a mechanism to measure the cognitive load during pre instruction, post instruction and in instructional stages of learning. This mechanism is used to help the instruction. As the load changes the instruction is made to adapt itself to ensure cognitive viability. This mechanism could be incorporated as a sub domain in the student model of various computer based instructional systems to facilitate the learning of programming.

**Keywords**—Cognitive load, Working memory, Cognitive Load measurement.

## I. INTRODUCTION

ONE of the key challenges that makes computer program learning tedious is cognitive load. This is mainly due to the nature of the complexity of computer programming and also due to the limited ability of the working memory to process information. In general cognitive load can be classified as two namely intrinsic and extraneous. Intrinsic load cannot be reduced much as it is fully dependent on the subject itself. But extraneous load can be reduced considerably by appropriate use of instructional design.

The working memory can be extended easily since the working memory theory states that working memory processes audio based information and visual based information separately. Capacity of the working memory is about seven chunks of information. For example if a seven digit number is presented to a learner the learner is able to assimilate and recall the number. Thus an appropriate use of visual aid could help in reducing cognitive load and based on

Muhammed Yousoof is with the Department of Computer Science College of Arts and Applied Sciences, Dhofar University, Salalah, Oman (e-mail: m\_youssoof@du.edu.om).

Mohd Sapiyan is with the Department of AI, Faculty of Computer Science, University of Malaya, Malaysia (e-mail: pian@um.edu.my)

Khaja Kamaluddin is with the department of Computer Science, Dhofar University, Salalah, Oman (e-mail: k\_kamaluddin@du.edu.om).

this assumption many software tools have been developed to visualize the program. But in spite of this independent nature of processing of audio and visual information there is a limitation on the leverage on the cognitive load.

But in the case if a larger number is represented then the learner could not remember the number and this is caused due to cognitive load. So it is inferred that

$$\text{Total cognitive load} = \text{Intrinsic Load} + \text{Extraneous Load} + \text{Germane Load}.$$

The cognitive load could be reduced by tailoring the instructional design, or extending the working memory by appropriate use of visual images to process the information. But the teacher's role can be only done on the extraneous load. If the load is peak then the instructional methodology should be tailored to facilitate learning.

## II. EFFORTS AND METHODS ADOPTED TO REDUCE COGNITIVE LOAD

Various efforts made to reduce cognitive load like software visualization [8],[9]. Another effort used to reduce cognitive Load is usage of the system like COLORS[10] which presents partial program code which in turn helps in reducing the load. This could be assumed on the basis that the information processing of visual and auditory information is done separately. It is also proved from previous research that split attention effect which could be caused by presenting the instruction in different modalities like text, audio and video is ruled out. Cognitive load needed for integration of information is different modalities is not required, since working memory has the ability to process the visual and auditory information independently.

As mentioned in the abstract various measures have been tried in previous researches to reduce cognitive load, but an appropriate measure to control the cognitive load was not adopted. The term cognitive load appears to be vague and is not implied properly in learning cycle. This causes the learner a difficulty. When we discuss about the measure of cognitive load many methods people have adopted namely analytical methods, secondary task techniques, rating scale techniques, physiological techniques have been tried. A literature survey on the study of cognitive load measurement has shown that starting from 1988 until 2002 most of the efforts to measure cognitive load are rating scale techniques [7]. There are few efforts made using physiological methods like heart beat, retina expansion etc[2]. There have been doubts over the reliability of the rating scale technique in measurement of

cognitive load. But most of the studies have proved that the people could share correctly their cognitive overload that happened during the process of learning[2]. Other methods adopted like heart beat variability, retina enlargement etc could not achieve the desired success. The success rates of the various methods adapted to measure the cognitive load has not yielded good results. For example when using enlargement of pupil as a measure to indicate the cognitive overload could not be precise since the pupil could be enlarged due to external factors like light, dust etc. Similarly as mentioned, most of the previous measures which used the rating scale asked the user to prompt a feedback on the cognitive load experienced which could add an additional burden on cognitive load experienced by the learner. Similarly the cognitive load measurement techniques like secondary task measurement could also not work well as it leads to additional load due to the secondary task added up in the instruction. Example of a secondary task could be worked examples[5]. Some efforts have been made to use Intelligent Systems which can tailor instruction as per student's performance where the student modelling is designed based on Bayesian Networks[4]. The Computer aided learning systems also segregated the students based on personality[1] and cognitive trait. Some systems have incorporated Cognitive Trait Model to ease the learning process [6]

### III. TAXONOMY OF THE PROPOSED MODEL TO MEASURE COGNITIVE LOAD

In this research we propose to use a computer based tutoring system which will have the ability to measure the cognitive load during pre instruction, in instruction stage and post instructional stages. Depending upon the cognitive load the system will modify the instructional design. The proposed model will have three main components namely student model, pedagogical module and user interface which are the elements of the normal intelligent tutoring system. The variation in our proposed model will in incorporation of the cognitive load measurement model to monitor the cognitive load and tailor the instruction as per the situation. The following diagram represents the proposed model.

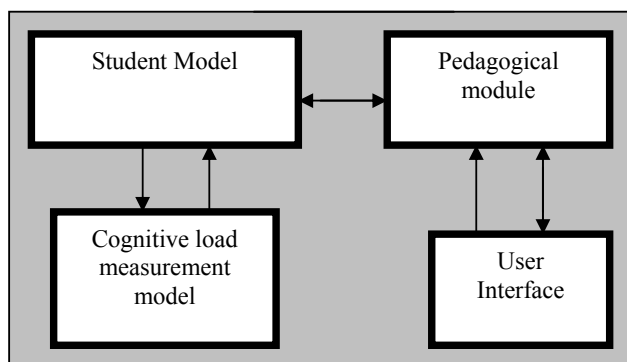


Fig. 1 Modified architecture of Tutoring system with Cognitive Load Measurement Module

The cognitive load measurement model will help measure the cognitive load. When an instruction is presented to the learner then model will break the instruction into chunks of

information. If the chunk of information is more than seven then it will communicate to the pedagogical module to alter the instruction. In short it can instigate the use of visual memory for instance if it finds some visual information can be processed. This is during the pre instructional stage.

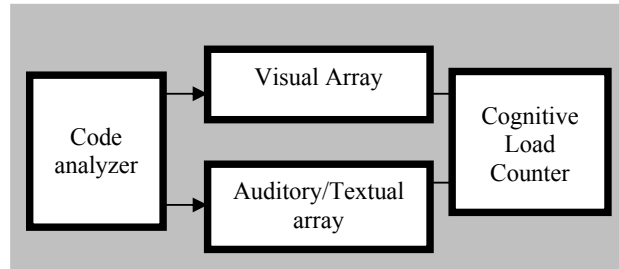


Fig. 2 Components of Cognitive Load Measurement Model

During the In instructional stage it will also monitor that the chunks of information presented for processing is seven or less than seven so that cognitive load is maintained. This mechanism which is embedded in the cognitive load measurement model will be implemented practically by using the concept of simple Array. Two arrays of maximum seven elements will be created; one array will be used to store the textual and audio type of chunks and another array to store the visual types of chunks. In this model two arrays are used since the visual information and auditory information are processed separately. This assumption is proved based on the working memory model proposed by previous researches. When ever an instruction is presented to the learner it will be broken down into chunks of information. This chunks of information will be stored in individual elements of the array. The storage will depend upon the nature of the information. If the information is visual based it will be added to the visual array and if it chunk is classified as auditory or textual type then it will be added to the text array. If the sum of the elements in both the arrays have exceeded seven then the cognitive load model concludes that cognitive overload has happened. In that case then the cognitive load measurement module communicates with the pedagogical module to either modify the instruction or extend the working memory depending upon the elements processed either textual or visual based information

In the proposed model when the program is presented to the learner or when the learners attempts to learn a new program, program is broken down into chunks of information by the component called code analyzer. The function of the code analyzer is to analyze the code and break it into chunks of information. Each chunk of information will be stored in the array either in visual array or auditory or textual array depending upon how the instruction is presented to the learner. There will be a built in mechanism which has a counter to keep track of the number of the chunks of information processed at a time by adding the sum of the elements in both visual and auditory/textual array. When the counter reaches seven at one point of time, it indicates that a maximum load is processed now during the learning process. If any information is not needed at any point of time it will be removed from the

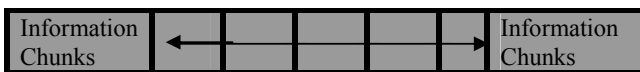
array just decrementing the number of elements in the array. Subsequently the cognitive load counter will be decremented. This mechanism using the array and counter will help in maintaining the cognitive load and if any excessive load happens it will modify the instruction so as to avoid overloading.

The following diagram illustrates the storage of information in the array to measure the cognitive load.

Visual Array V[0] to V[6]



Textual Array T[0] to T [6]



The cognitive load model also measures the impact of the mental load during the learning process by analysing the performance. At the end of the instruction a short quiz is presented to the learner and depending upon the response of the learner, the module could infer whether the cognitive load has happened or not. For instance if the learner is presented a quiz of five questions and if all answered correctly then it shows learning has been achieved and no cognitive load happened in the process of learning. Depending upon the number of wrong responses the cognitive load model could initiate remedial measures to be taken by pedagogical module.

#### IV. COGNITIVE LOAD REDUCTION IN THE FRAMEWORK

The proposed framework will try to overcome the two difficulty areas namely misconceptions of concepts and reducing cognitive load. A good learning support will help to overcome this problem. The proposed visualization will give the learning support appropriate for each individual learner. So it will result in ease of learning programming.

Many learners are unable to understand many concepts used in programming for the simple reason they are abstract. For all types of learners learning support should be given to understand the concepts. For examples abstract concepts like data encapsulation, information hiding etc. can be well understood with the visualization of concepts using simple examples. This approach leads to reduce cognitive load and also retain the ideas clearly.

The current approaches to visualization have tried to use visual metaphors to illustrate the different areas of program. In some systems the animation of the program is shown. But in the proposed framework visualization of the program visualization can be done in many ways like UML notations, Concept Maps, Partial coding, Program reading etc. This multicentred approach in visualization will cater to each individual user's ability. This can help the learners to progress by their own phase and style. In short customizing the visual support on the basis of cognitive trait is the strength of the system.

#### V. PROPOSED ALGORITHM TO REDUCE COGNITIVE LOAD

*Begin*

*Code Analyzer analyzes the program code or instruction presented for processing*

*Instruction presented is divided into chunks of information*

*CogLoad = 0*

*T [7] = { }*

*V [7] = { }*

*If information is presented in auditory/textual format then*

*Store the information to Text Array*

*T[7] = [value1],[value2] .....[value 7]*

*Else*

*If information is presented or processed in visual format then*

*Store the information to Visual array*

*V [7] = [value1],[value2] .....[value 7]*

*For every element added in text or visual array*

*CogLoad = CogLoad +1*

*If CogLoad >= 7 then*

*"Indicate Cognitive Load has attained maximum limit"*

*reorganize the instruction by communicating with pedagogical module*

*If any information is no more needed for processing*

*Delete the information from the array*

*For every deletion from the array*

*CogLoad = CogLoad -1*

#### VI. DISCUSSION AND FUTURE WORK

The above model of measuring the cognitive load during the Pre Instructional, In-Instructional and Post Instructional stages could help determine the overload experienced by the learner. When overload has reached the pedagogical module will be informed to alter the instruction so as to reduce the load. In computers for example the working memory could be expanded by adding additional RAM so as to process more information. This is not possible with the human brain as its limit is limited to only process maximum seven chunks of information. With this limitation, the only option left is to utilize the available working memory efficiently. The above proposed model of measuring the load by dividing the code into chunks of information by code analyzer and subsequently try to store the instruction in visual and audio/text array will improve as the visual and text information are processed separately by the brain. The various other measures that has been used to monitor the cognitive load such as self evaluative assessment, physiological measures etc. had its own limitations. This proposed system is a modification to the existing intelligent tutoring system with an addition of a module called Cognitive load measurement model which has intelligent element to monitor the load and inform the other modules of intelligent tutoring system to tailor the instruction when cognitive load happens.

# REFERENCES

- [1] Amy.B et.al (2004)."Personality as a predictor of student success in programming principles". Proceedings of the 7<sup>th</sup> Annual conference of Southern Association for informing systems.
- [2] C. Aschwanden & J.Stelovsky (2002). "Measuring Cognitive Load with EventStream Software Framework". Proceedings of the 36<sup>th</sup> International Conference on System Sciences (HICCSS 03), Hawaii, USA.
- [3] Carlos Miguel Tobar et.al(2006)." An explanation Tool to Support Learning of Basic Programming". Proceedings of the 36<sup>th</sup> ASEE/IEEE Frontiers in Education conference, San Diego, USA.
- [4] Christine Lee & M.Sapiyan Baba (2005)." Work in Progress – Development of a 3-tier Architecture C++ STL ITS.Proceedings of the 35<sup>th</sup> ASEE/IEEE Frontiers in Education conference, Indianapolis, USA.
- [5] Fred Paas et.al(2003)."Cognitive Load Measurement as a Means to Advance cognitive Load Theory". Journal of Educational Psychologist,38(1), 63-71.
- [6] Lin..T., Kinshuk & Patel.A(2003).Cognitive Trait Model- A supplement to Performance Based Student Models.Proceedings of the International conference on Computers in Education,Hong Kong.
- [7] Luz M.Quiroga et.al (2004).Reducing Cognitive Load.Proceedings of the 37<sup>th</sup> International conference on system sciences, Hawaii, USA.
- [8] Maria Satratzemi et. al (2001). A system for program visualization and problem solving path assessment of novice programmers. Proceedings of Annual Joint Conference on Integrating Technology in Computer Science Education, Page 137 -140, Canterbury, UK.
- [9] Ronit. B & et. al.(2000).An Extended Experiment with Jeliot 2000.Proceedings of the Program visualization workshop,University of Joensuu, Finland.
- [10] Stuart Garner (2002)," COLORS for Programming: A System to Support the Learning of Programming", Proceedings of Informing Science.