World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:1, No:12, 2007

Reducing Cognitive Load in Learning Computer Programming

Muhammed Yousoof, Mohd Sapiyan, and Khaja Kamaluddin

***Abstract*—**Many difficulties are faced in the process of learning computer programming. This paper will propose a system framework intended to reduce cognitive load in learning programming. In first section focus is given on the process of learning and the shortcomings of the current approaches to learning programming. Finally the proposed prototype is suggested along with the justification of the prototype. In the proposed prototype the concept map is used as visualization metaphor. Concept maps are similar to the mental schema in long term memory and hence it can reduce cognitive load well. In addition other method such as part code method is also proposed in this framework to can reduce cognitive load.

***Keywords*—**Cognitive load, concept maps, working memory, split attention effect, partial code programs.

I. Introduction

PROGRAMMING is often considered a difficult task due to the amount of complexity involved in it.Many computer scientists could not cope up with acquiring the skills of programming even after the completion of the course.

Previous researches indicate that the reason for non achieved learning outcomes may be due to

1. Complexity of programming language syntax and the concepts
2. Cognitive load involved in learning programming
3. Poorly designed instructional methodology
4. Misconceptions of the concepts

As teachers we realized the core difficulty area to ease overcome learning difficulty in programming is to manage the cognitive load in a systematic way. If cognitive load is managed well in program learning then all other difficulties can be overcome.We realized the all the problems related to learning programming is the offshoot of the cognitive load involved in programming.The problems stated above results in the failure of meeting learning outcomes.

II. Process of Learning

Learning process involves the usage of memory. There are three types of memory namely long term memory, sensory memory and working memory. Each memory has its role in any learning process.

Muhammed Yousoof is with the Department of Computer Science, Dhofar University, Salalah Sultanate of, Oman,. (e-mail: m_yousoof@du.edu..om).
Mohd Sapiyan is with the Faculty of Computer Science and Information Technology,University of Malaya, Malaysia(email:pian@um.edu.my)
Khaja Kamaluddin is with the Department of Computer Science, Dhofar University, Salalah, Sultanate of Oman. (e-mail: k_kamaluddin@)du.edu.om)

Long term memory- This memory stores the information in more or less permanent way so that the information stored can be recalled during the process of learning.The long term memory is responsible for building schemas or mental image.Schema is a network of information about any thing. For example when we think about programming, a schema can be like this as shown below:

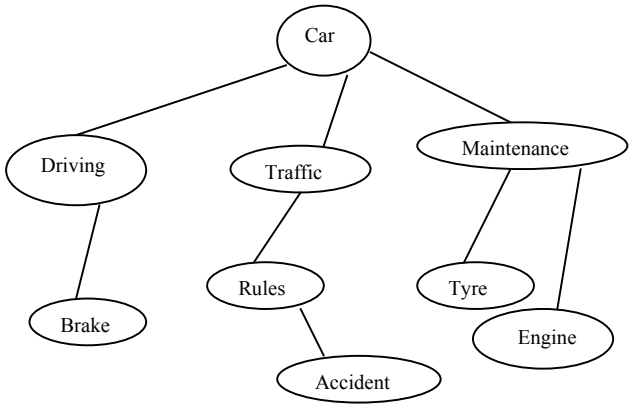

Fig. 1 Mental Schema developed in LTM

The schema differs from person to person .For example the schema of a novice programmer is not complex as that of an expert. So novice programmer has to consume more time in solving a problem as the schema in the long term memory is not available. In short the novice lacks information about the concept that he /she learns and it cannot be recalled from long term memory. Long term memory classifies the person as novice or an expert.If the schema is complex and expanded then that person can solve task in that domain with ease.

Working memory- This memory is very limited .This memory actually process the information to solve the problems. Previous researches have found that the working memory is limited to seven elements only at a time. If at any point of time more than seven elements need to be processed at the same time, it leads to over load .When overload occurs then learning outcome cannot be achieved.

Senory memory: This is short memory .It is the stimulus received by the learner through sensory organs like eyes, ear, touch, smell, taste. The stimuli received from the senses are perceived by this memory which is processed in the work memory.

The typical model of memory associated in learning is as shown in Fig. 2.

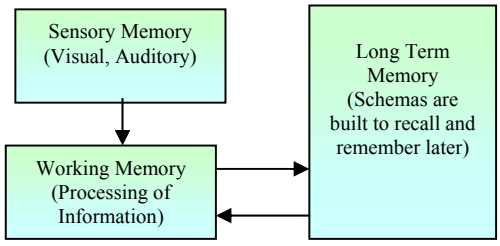

Fig. 2 Model of Memory involved in learning programming

III. Role of Cognition in Learning Programming

There are various levels of activity to cognition in programming. Programming is composed of elements which are highly interactive and this leads to cognitive load in working memory. The intrinsic cognitive load is high for certain concepts in teaching programming.But extraneous cognitive load can be reduced by restructuring the instructional design, so that the load on working memory can be reduced or enhance the working memory by utilizing sensory memory appropriately.Utlization of sensory memory may be presenting pictorial representations, which can help extend the working memory as picture presented may not need processing by the working memory.

There are many skills to be needed at one instant or at different instances in the process of learning programming. Some times the skills needed at one point may exceed seven elements which is the threshold of working memory.This leads to information loss and thereby learning outcomes may not be met.The following tables summarizes the set of possible cognitive tasks involved in learning programming.

TABLE I
Levels of Cognitive Skills Needed for Programming

Sensory	Vision, Auditory
Subconsious	Memory, Pereception, Motivation, Goal setting
Meta cognitive	Attention Concept formation, Abstraction
Categorization	Knowledge representation
Higher Cognitive	Problem solving Reasoning, Decesion making, comprehension

IV. Current Approaches to Learning Computer Programming

Most of the curriculum and instructional designers have focused to develop programming curriculum in the hierarchy of learning difficulty. Mostly the students are taught the concepts in lectures with worked examples. The usage of visual aids like pictorial representation of the program using flow chart or UML diagrams in the case of Object oriented programming is supplemented in the lectures. Then the learners are given practical a session by giving exercises to practice and assimilate the concepts clearly.

This approach has not been effective as many novice programmers find it difficult to acquire the ideas completely.It sometimes leads to mere memorization of programming codes to score marks in examinations. Later in practical work place the computer science graduates cannot code programs.

This approach may lead to split attention effect as students when learning computer programming in labs have to rely on many sources like tutorials,lecture notes and practicals sessions.During this process working memory is utilized to integrate multiple resources and thereby working memory is not fully utilized to solve problems or learn programming.

In the process of learning computer programming much information is presented both textually and pictorically.Many believe that introducing pictures along with textual information will foster learning.But on the other hand this may lead to cognitive load as more mental resources are utilized to process the information from multiple resources rather than single source either textual or pictorial.

V. Innovative Approaches to Learning Computer Programming

According to [9] it is proposed that cognitive overload can be reduced by the use of Category-Avoiding examples.In this approach to learn a new thing, a modular example emphasizing on the part of the bigger problem should be presented to allow the learner to assimilate the idea. Normally the learners are presented with the category focusing examples, which is presenting complete worked examples.This category focusing examples can increase workload.

According to [10] cognitive load in learning programming can be reduced considerably by providing learning supports like the part complete solutions of the program. This proposal is based on the Oliver's instructional design model.

According to [2] to improve the style of programming, pair programming is suggested. In pair programming one learner is observer and other one is developer. In[12] learning programming can be made effective by including immersion method, which is based on reading programs to understand and then try to write program on own.

VI. Concept Map as Metaphor for Reducing Cognition in Learning Programming

Based on the current situations we propose to reduce cognitive load by visualization of programs.Program visualization is to show visually the run time behaviour of the program during run time in many visualization sytems.. The key considerations for this framework will be to integerate learning support appropriately in the form of single source so as to utilize working memory fully and to avoid effects like redundancy and split attention. effect. This framework will also use the idea of expanding the working memory by proper use of visual senses, as previous researches have illustrated

that part of the working memory is allocated to process visual information perceived from sensory memory. Thus working memory can be utilized in the best way.

When discussing about visualization the first and foremost thing that comes to mind is the use of metaphor to show the visualization of program.There are two types of metaphors namely visual metaphor and verbal metaphor.Among these visual metaphor is better due to the reason that visual sense can retain and help process information faster.

The following table illustrates the metaphors used in various visualization tools.

Visualization System	Metaphors used
Jeliot	Theater
BlueJ	UML like Class diagrams
Plan Ani	Role based metaphors

In our proposed model we propose to use concept maps as the metaphor to visualize the programs.Concept maps are very useful tools to determine the various aspects of a concept and the relationships between the various concepts. Concept map structure is similar to schema structure in long term memory. As discussed in the previous section the complexity of the schema of each individual determines the level of difficulty in understanding and solving the problem. So assimilation of the concept map will foster in building up the schema in the process of learning. If the schema is complex for the individual then the need of processing information in the working memory becomes less which results in cognitive load. As concept map is analogous with the mental schema in Long Term memory, it can reduce to the load significantly.

Another advantage of using concept map is it requires explication and reflection and may help pupil to develop auto monitoring techniques and so enhance their critical thinking. In addition to this during the construction of concept maps the learner not only acquires knowledge and gaps in learning process, but also the individual learning strategies. This can help learners to correct or enhance learning by correcting the learning strategies[4]

VII. Prototype of Proposed Systemms for Reducing Cognition

To explain the proposed prototype we consider the following program written in Pascal

```
program palindrome;
const MaxLeng =8;
var len,i: integer
candidate: array[1..MaxLeng] of char;
palindrome: Boolean;

begin
  repeat
   write('Enter the Length of the string')
   readln(len)
    if( len <1) or ( len > MaxLeng
       then writeln('Must be between 1 and MaxLen)

until (len>=1) and (len<=MaxLeng;
for i= 1 to len do begin
write('Enter', I'.letter:');
readln(candidate([i])
End;

palindrome :=true
for  I := 1 to len do

palindrome := pali and (candidate[i]]=candidate[len+1]);

if palindrome

then write('It is')
else write('It is not');
writeln('a palindrome')

end
```

In our proposed prototype as the user types the program in the program editor window, concept map for that particular program will be constructed simultaneously to visualize the program.This visualization is static is nature .A sample prototype is as shown below:

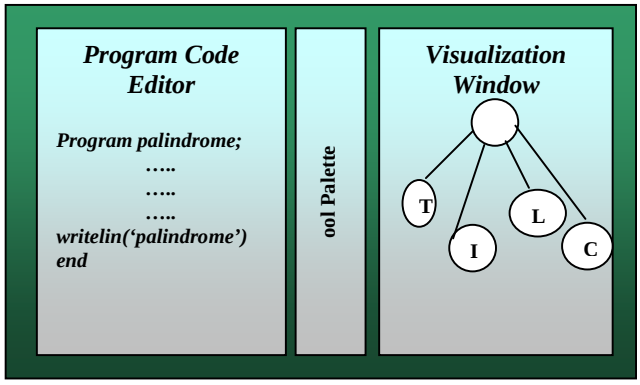

Fig. 3 Proposed system's prototype

In the visualization window the concept map for the program is constructed in parallel as the learner starts writing the program. The visualization window will have the four nodes in the initial stage namely constants, variables, input/output,loops. As the program expands the related information will be attached to the corresponding nodes.For example in the above program the variable len and i will be attached as nodes in the concept map then the link is established between the variables node and the two new nodes namely len and i Similarly the other nodes are also expanded as the program is written in the program editor. When the program writing is finished the complete concept tree for the program is displayed.

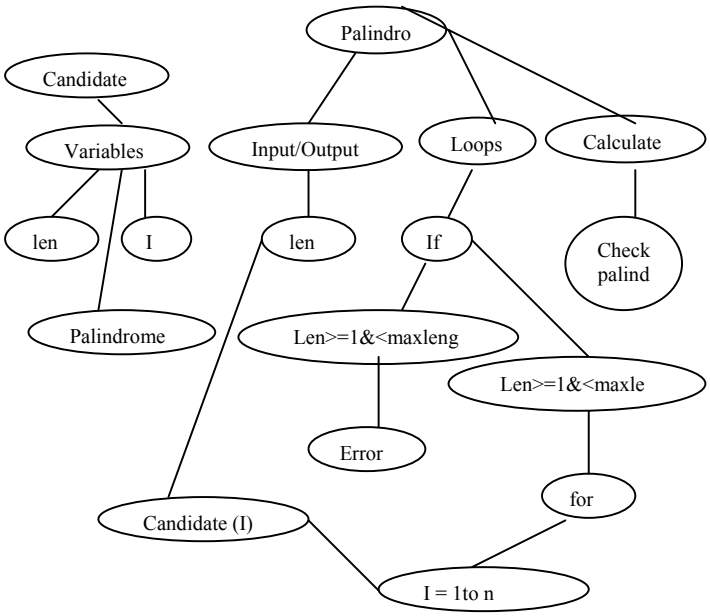

Fig. 4 Visualization of the program using the Concept Map

In the tool palette we propose to have five buttons namely variables, loops, Input/Output, constants and Partial code. When we click on the variable button the variable declared and used in the program alone will be displayed in the program editor window and simultaneously the learner will only part of the concept tree which will show the nodes under the variable node and the relationship between them. The other buttons namely constants,loops,Input/Output will also function in a similar way. This approach can also reduce the cognitive load as information filtering takes place here. The user can view only the limited information necessary to concentrate at a point of time.

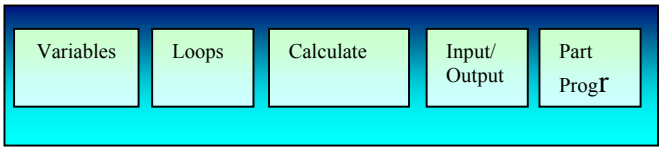

Fig. 5 Tool palette of the proposed system

The purpose of having "PartProgram" feature is to use present the learner the support through a single source namely text.When this button is clicked the program is loaded with a new Girths GUI will have two windows one where the partial code is presented and the other where the support is given to learner to learn in the form of some possible missing codes. The learner will drag and drop the codes to the partial code and if correct the learner will be allowed to precede otherwise error message will be displayed to guide the learner.

A sample prototype is as shown below.

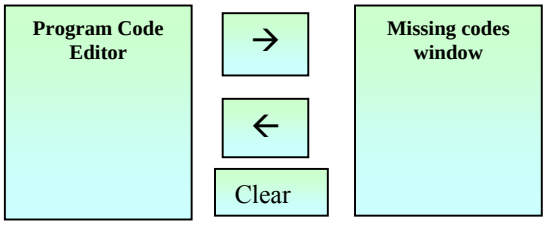

Fig. 6 Part program approach to reduce cognitive load

VIII. Discussion and Future Work

There are many attempts tried out to ease the process of learning programming. Cognitive load which is the core difficulty area that needs to be addressed to facilitate learning programming.The proposed framework will help in reducing the cognitive load as we have proposed to use concept maps as the visualization metaphors which can instigate building the schema in the long term memory. We have also proposed to use information filtering in the visualization process thereby the working memory can be less burdened and the program is presented in smaller chunks so that the learner can assimilate each part.Grouping of information on the basis of a taxonomy will help to foster learning.

The system will be developed on the basis of the framework and then the effectiveness of this framework can be studied through analyzing student's performance after using the proposed system.

References

[1] Amy.B et.al (2004)."Personality as a predictor of student success in programming principles". Proceedings of the 7th Annual conference of Southern Association for Informing systems.

[2] Alistair CockBurn & Laurie Willams .The cost and benefits of pair programming. Unpublished Paper on Pair programming.

[3] Clark Ian et.al (2004).Using concept to plan an introductory geology course.Journal of Geoscience Education.May 2004 Issue

[4] Eric Bruilliard & Georges.L.Computer Based Concept mapping- A review of cognitive tool for students. Proceedings of ICEUT 2000 pages 331-338.

[5] In J.Kuljis,L.Baldwin & R.Scoble.Programming aptitude testing as a prediction of learning to program. Proceedings of the 14th Workshop of the psychology of programming interest group.Brunel University,UK

[6] Kinshuk&Lin T(2004).Cognitive profiling towards Formal Adaptive Technologies in Web Based Learning Communities. International Journal of www-based Communities,1(1)103-108(ISSN 1477-8394).

[7] Lin T.,Kinshuk&Patel.A.(2003).Cognitive Trait Model-A supplement to Performance Based Student Models. Proceeding of the International Conference on Computers in Education 2003,Pages 629-632.

[8] Luz M.Quiroga et. al(2004).Reducing Cognitive Load". Proceedings of the 37th Hawaii International conference of systems sciences.

[9] Peter Gerets et.al(2003).Reducing Cognitive Load and Fostering Accquistion examples-Benefits of Category Avoiding Instructional Examples.Proceeding of International Conference.Pages 450-455.

[10] Stuart Garner(2001)." A tool to support the use of part complete solutions in the learning of of Programming". Proceedings of Informing science-2001.

[11] Vekiri,I,&Samson P.(2000).Applying Cognitive Research to the Design of Visualization Tools: Features of Blue Skies-College Edition Software.Proceedings of the Fourth International Conference of the Learning Sciences(pp 106-107).

[12] William Campbell &Ethan Bolker(2002)."Teaching programming by Immersion,Reading and writing".Proceedings of 32nd ASEE/IEEE Frontiers in Education conference.

[13] Ying Xu Wang et.al(2003)." A Layed Reference Model of the Brain". Proceedings of the 2nd International Conference on Cognitive informatics(ICCI'03).