Performance Evaluation of the Software Visualization tools and A New Framework to Manage Cognitive Load in Computer Program Learning

Muhammed Yousoof,
Mathematics & IT Unit,
Dhofar University,
Salalah,
OMAN,
m_yousoof@du.edu.om

Mohd Sapiyan
Faculty of Computer Science & IT
University of Malaya
Kuala Lumpur
MALAYSIA
pian@um.edu.my

K.Ramasamy
Elec & Computer Eng Dept
Dhofar University
Salalah
OMAN
k.ramasamy@du.edu.om

Abstract: - Cognitive load experienced while learning programming is very high due to the high element of interactivity and poor instructional design. Prior researchers [2][5][6][10] have focused to minimize the load such as program visualization, pair programming etc. There are many computer based tools available in the market which are aimed to reduce the cognitive load experienced by the learners and to facilitate faster learning. In this paper, two such tools such as JELIOT, BlueJ are taken into consideration. We evaluate the effectiveness of each of the above mentioned tools in their ability of easing the learning process. An experiment was conducted among the students of Computer Science at Dhofar University to evaluate the effectiveness of each tool to reduce the load experienced by the learner. The impact of these measures is not determinable since there is no mechanism to monitor the load and thus the results of the previous studies are very subjective. We also propose a framework which consists of 3 layers, to help in managing the load by monitoring .When the load exceeds the capacity, the instructional design could be altered or customized to enable the learning. The proposed framework is a novel way to ease the learning process of computer programming.

Key-Words: -Cognitive Load, Jeliot, BlueJ, Physiological Measures, Galvanic Skin Response.

1 Introduction

Computer Programming is considered as a difficult task by many learners. This is due to the fact lot of challenges are faced in the learning process. Many learners feel it is complex and highly cognitive. There are many problems associated with learning programming but as an academic teaching computer science, it is our perception based on our experience that the real problem is cognitive load.This is augmented by the available literature[12][13][16].

Many efforts have been done to reduce the cognitive load in learning programming[6][11] Most of the efforts are aimed to reduce extraneous load by appropriate methodology of instruction One such effort is the usage of Intelligent Tutoring Systems which is an offshoot of Computer Based Instruction(CBT).There are a variety of Intelligent Tutoring systems to teach programming and to name a few JITS[14], .The goal of Intelligent Tutoring systems is to customize the instruction intelligently by using AI techniques like Bayesian Networks, Fuzzy logic to design the student model and pedagogical to suit the learning styles of each individual learners.

Another Effort to help the learners is the use of software visualization techniques. It is augmented that load could be reduced easily when the instruction is presented in visual and text/auditory format simultaneously. This argument is based on the Baddeley theory on working memory [1]. Some of the available visualization systems are BlueJ [4] which uses the visual metaphor similar to UML diagrams. Jeliot which used the theatre metaphor [10]. Another system called COLOR [12] is used and it is based on the Oliver's Learning Model which includes three activities namely learning activities, learning support and learning resources. CORT (Code Restructuring tool) [12] is also used to reduce the load by presenting the partial codes and the learner has to fix the correct codes. Pair Programming is considered as a technique to reduce the load as it involves the group effort. Some efforts to reduce the cognitive load included controlling exploration space [9]. Other efforts include the idea of Cognitive Trait Model [7]

All the above mentioned efforts have aimed to reduce the load. But there is no solid proof on how effective the above mentioned mechanisms or methods have reduced the load. Various studies

were done to study the effectiveness of such system. But it is observed that a load measurement component is missing whereby it is not easy to predict when the load happens. So we propose a mechanism to measure the load while learning programming, which helps us to determine the load level experienced and depending upon the load experience at that time, instructions can be customized or optimized. This approach will help the learners in easing the barriers faced in learning programming. So in this research the measuring of load is integrated thereby helping us to clearly monitor the load and thus manage the instruction efficiently.

The paper is organized as follows: In Section 2, the effectiveness of the current visualization systems to reduce the load is discussed, which includes the experimental setup and results. In Section 3, a new framework for managing cognitive load is introduced. In section 4, Cognitive Load Measurement techniques which include a discussion on both physiological and non physiological methods are provided, In section 5, optimizing the instruction is discussed. Conclusion and future work is provided in section 6.

2. Current visualization systems

A key area of research to help the ease of programming is software visualization where the program is animated and thus enables the learners to visualize the concepts of programming. It is believed that the information presented in the visual media can be assimilated very easily. This is well accepted by the fact that the teachers of all disciplines have been insisted to use visual aids in the teaching process. Based on the available literature there are many systems available to visualize the programs and such as BlueJ [4] which is used to teach Java Programming, Jeliot [10] which is used to teach Java programming. Another system is COLORS/CORT [3] which uses the idea of partial code presentation and thereby reducing the load of learner. In this section the main focus will be on evaluating the effectiveness of the above mentioned systems in reducing the cognitive load of novice programmers. In the final section will propose ways to improve the existing systems.

2.1 Description of the current systems

The two systems that are taken into discussion namely , BlueJ and Jeliot are used to teach elementary programming .BlueJ uses UML notations as metaphor. Jeliot uses theatre metaphor.

2.1.1 Blue J

BlueJ visualization system uses the concept of UML diagrams to animate the program. The program is animated in terms of class diagrams and other standard notations used in the Unified Modeling Language.

The sample screen shot of BlueJ is as shown in the Figure 1 .

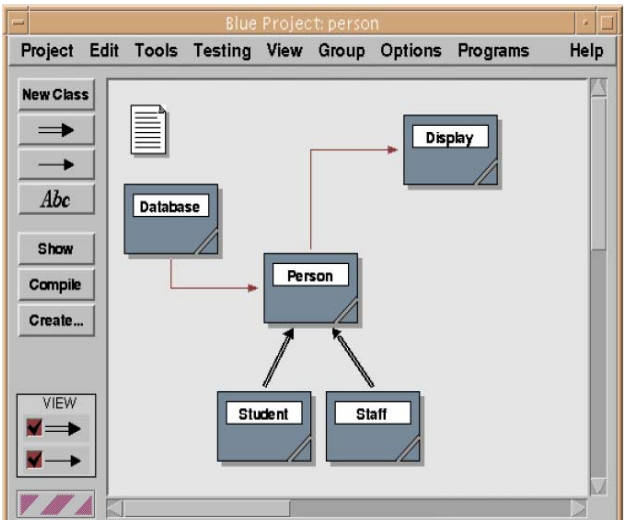

Figure 1: Screenshot of Blue J

2.1.2 Jeliot

The sample screen of the Jeliot system is as shown in Figure 2

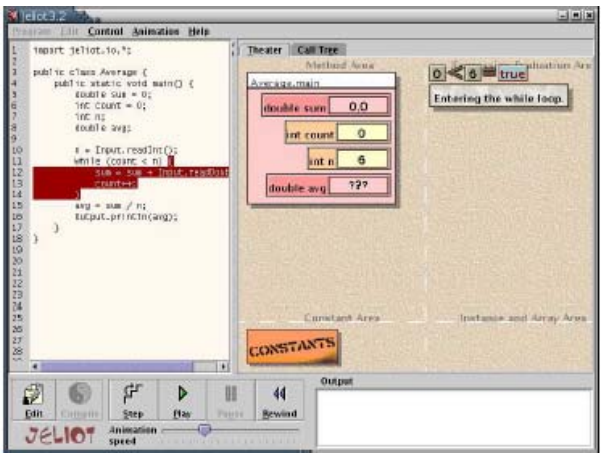

Figure 2: Screenshot of Jeliot

In Jeliot system each element of a program is considered as an actor. A user has an option to animate the program step by step or animate incrementally without user interaction. There is also a provision to rewind the animation process.

2.2. Performance evaluation

The two systems discussed in section 2.1 have contributed helping the learning process for beginners. But we are interested in the evaluating how much effective each of this system in reducing the load.

We have choosen nine students, who are doing graduate program in computer science at Dhofar University, Oman and we divided them into 3 groups namely Jeliot, BlueJ and other a Regular Group. Each group is given the same program to learn, but with different visualization systems and the last group was mad to learn programming without any visualization system.

We evaluated the effectiveness of the system using the efficiency of understanding the concept by the learners. The extent of understanding of the concept is checked by an interview between the learner and the lecturer after the exercise. Usability tests were conducted by giving a questionnaire to evaluate the user friendliness of the system.

The learning outcomes of each learner are evaluated at the end of the activity and the extent of the learning outcome is evaluated based on the performance .A short quiz was also conducted to evaluate the extent of understanding of the concepts.

2.3. Results and discussion

The outcome of the experiment is provided in Table 1 and Figure 3. Table 1 provides the information on the usability of the various systems for learning. It is found that almost both the system namely BlueJ and Jeliot found equally efficient. The maximum difficulty was faced by students who were not exposed to any visualization environment.

	Time Taken for learning in Minutes		
	Easy Program	Average Program	Difficult Program
Jeliot	4.33	8	10
BlueJ	4.67	7	11
Regular	8	10	13.33

Table 1: Ease of learning using various visualization systems

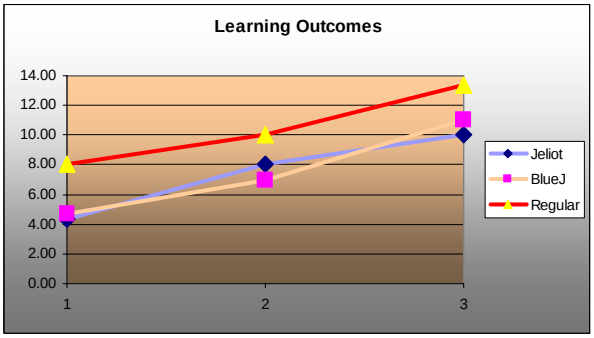

Figure 3: Ease of learning using various visualization systems.

Table 2 and Figure 4 shows the level of learning outcome achieved by using various visualization tools. It is observed that the use of visualization systems could yield approximately same results. The regular group could not meet the outcomes as achieved by BlueJ and Jeliot Group.

Level of Learning Outcome achieved			
	Easy Program	Average Program	Difficult Program
Jeliot	100	70	80
Blue J	90	70	80
Regular	75	50	30

Table 2: Learning outcomes vs various visualization tools

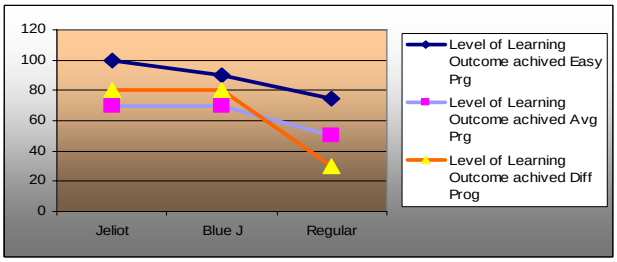

Figure 4: Results of various visualization tools vs learning outcomes

From the above study, it is inferred that certainly any visualization environment helps in reducing the load. But the most certain issue that needs to be addressed is individual differences in learning styles and learning phase. It is found that the students in the same group did not experience the same difficulty level or same type of problems during the learning process. All the systems in use have tried to reduce the load but failed to pinpoint and monitor the load of a learner during the learning process. In that case the systems that are in use could not really help individual cases but try to generalize the student's level and understanding. This could lead to non learning as we ourselves find different students have different levels of skills and understanding ability. Therefore we propose a new framework where a new component will measure the load during the learning process and on finding the load has reached the threshold, the instruction will be customized to tailor individual learner's style. The customization or optimization of the instruction will be done using intelligent agents in the system or using optimization techniques. A detailed explanation and the theoretical framework on the need of a new framework is discussed in the subsequent sections.

3 New framework to manage the cognitive load

The proposed framework shown in Figure 5 consists of a three tier in which the three steps of learning are addressed which pre learning, during learning stage and post learning. There are three modules in the framework namely Load measurement module, Tutor Module, Performance Module. Each tier has its own functions and roles. The proposed framework is discussed in detail in the subsequent sections.

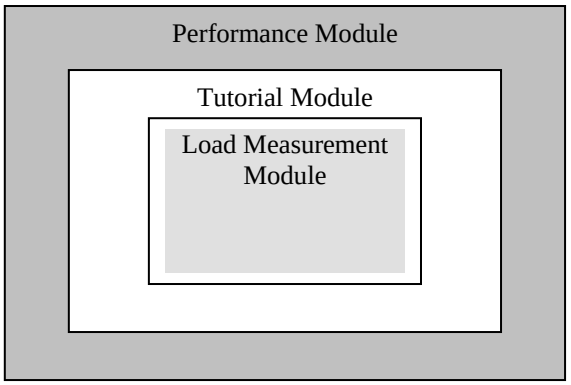

Figure 5: New framework for managing cognitive load

The first layer is called Load Measurement module, which will device a mechanism to monitor the load during the two stages of learning namely learning phase and pre learning phase. The second layer is tutor module which is designed to deal with the presentation of information to the learner in a dynamic way based on the input received the load measurement module. The instruction presented to the learner will be customized according to the load experienced during the various stages of learning.

The third layer is designed to deal with post instruction issue to verify the impact of load during learning stage and to validate the learning outcomes.

4 Measurement of cognitive load

In this section, the various techniques to measure the load namely physiological and non physiological measures are discussed

4.1 Non physiological measures

Most of the computer based systems have used rating scale [4][8][17]as a means to manage the load.In each user interface the user is asked to fill in the nature of experience he faced during the learning. Previous researchers have used a 9 point rating scale or seven point rating scale in most of the cases. Based on the score given by the user the cognitive load could be deteremined.But always this method has a limitation of biasness whereby the learner may present erroneous information about his experience of learning.
Another method often used to measure the load is presenting a secondary task along with the primary task [4]. The performance in the secondary task is often considered as the measure of load experienced by the learner. The load measurement is indirectly done by taking into consideration the difficulty level faced in the secondary task.

These non physiological measures have been used very widely in load measurement techniques. Researchers conclude them to be more accurate inspite of the fact that they are very subjective and depend on the feedback of the user.

An experiment was conducted at Dhofar University, Oman to study the impact of the Cognitive load using rating scale and physiological measures. The experiment involved 10 subjects in which 4 male and 6 female participants studying their degree program in computer science took part.

Each subject was given three different programs of varying levels which represent the three levels of difficulty namely easy, average and difficult. Each subject were asked to interpret the program code and the function of the program. For each task, the learners are asked to rate the difficulty and the time taken to complete the task was recorded. The results are further cross verified using secondary task method .In this case we used the ability of the student to interpret the given code for each program.The results are tabulated are as shown below.

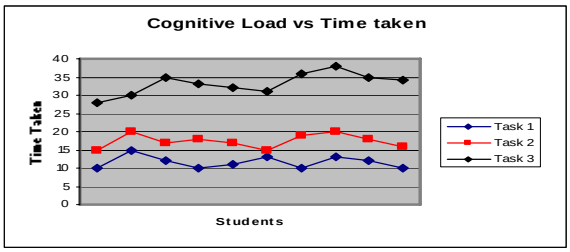

Figure 6: Time Taken vs Cognitive Load

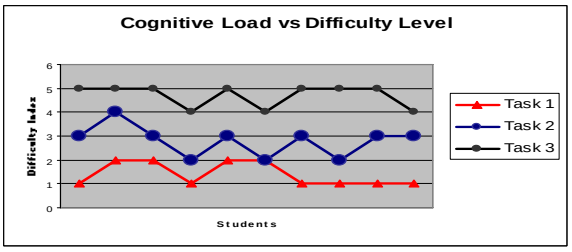

Figure 7: Difficulty level for the three tasks

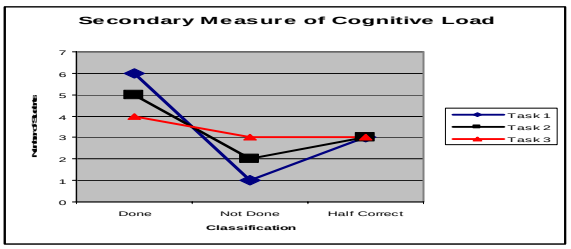

Figure 8: Secondary Task measure for cognitive load

The results are as follows. The cognitive load and the time taken for the three different tasks are provided in Fig 6. It is found that the task 1 took lesser time as it was easy where as the task 3 took more time for completion. Hence it is inferred that the time taken is an index of cognitive load experienced during the process of learning. The cognitive load and the difficulty level for the three tasks are provided in Figure 7 .The output of the difficulty level is got by the rating scale given by the students. In this study we used a likert scale of 5. Most of the students rated difficulty high for task 3 and where as task 2 as average and task 1 as easy. It is inferred that the cognitive load increases as the difficulty level increases. The figure 8 explains the relationship between the secondary measure and cognitive load. Task 1 which is easy most of the students were able to interpret the code indicating that load was not high and while on contrary the cognitive load was high for task 3.

4.2 Physiological measures

Another Common measures that the researchers have used to measure the load is using the physiological responses of the body.There are many ways adopted to measure the cognitive load in any task. Cognitive load can be simply defined as the mental effort needed to process given information.

Some researchers have tried to use physiological factors like heart beat rate variation, pupil size enlargement [13], perspiration rate variation in skin (Galvanic Skin Response) [17], temperature and pressure exerted on the mouse [17] The key problem encountered with this is that it is less feasible to implement this type of systems in a larger scale due to the nature of environment or devices needed. It also can give a wrong impression on the load experienced.For example, if the pupil is enlarged, people think the load experienced by the learner is huge.but on the contrary this may be due to the lighting in the class room etc.

In this framework, we propose to use the physiological measure of Galvanic skin response (GSR) as a means to measure the load. Prior efforts to reduce the load did not have an option of load measurement technique while learning programming. It is dynamic and in the sense that the variations of the load throughout the process can be monitored well and accurately the load can be measured . The impact of the external environment is also negligible in case of using GSR. In other physiological measure the external factors affect very much. We intend to use the hardware device PromComp Infiniti system of Thoughtechnolgies to measure the GSR variation when performing a task.

4.3 Basic assumptions for the terms used in measuring cognitive load

In this section, we derive mathematical formulations which can be followed up by developing an algorithm for this mechanism.

Cognitive Load: It is defined as the mental effort used to process any given information

CL = ME which is derived from TC/LC…(1)

where TC = *Task Character* which represents task characters such complexity, format of the instruction and time pace of the instruction.

and LC =Learner's learning character which could be Expertise level/ Age/ Spatial Ability.

Peak Load:It is defined as the threshold limit of the load that could be processed by the learner

PL = Max (Extraneous Load)…… (2)

Instantaneous Load : It is defined as the load experienced by the learner at any one particular instant

$$IL I = \int_{0}^{7} ME \ldots\ldots\ldots\ldots (3)$$

where IL is the Instantaneous Load at any given point of time.
Lower limit = 0 (minimum load)
Upper Limit = 7 (maximum load)
ME = Mental Effort

Accumulated Load:
It is defined as the sum of the instantaneous load experienced for any task in specific time quantums.

$AL = \sum(IL_1, IL_2, IL_3, IL_4, \ldots\ldots\ldots IL_n)$……..(4)

where AL = Accumulated Load
$IL_1, IL_2, IL_3 \ldots IL_N$ = Instantaneous Load experienced at different quantums (1,2,3,……..n are the different quantums)

Average load: It is the mean of the instantaneous load experienced during the learning of a particular task.

$ML = Average(IL_1, IL_2, IL_3, IL_4, ..IL_n)$…....(5)

where ML = Average Load for any task

IL_1, IL_2 = Instantaneous Load at any point of time for a given quantum of time limit.

Free capacity: It is the free space available In the working memory at any particular instant in any learning task.

$FC = PL - IL_i$.....(6)

where FC = Free Capacity available for usage in memory

IL_I = Instantaneous Load at any instant **i**

Based on the above definitions we now derive an algorithm that will help in managing the load while learning process takes place.

Algorithm:

The algorithm used to measure the cognitive load is given below

Set Peak load = 7

Set Instantaneous Load =0

Set Time Start = 0

Set Threshold time limit = 5 mins

Set Chunk counter =0

Divide the Instruction into tasks (chunks)

Learning Phase

Start the Instruction
Do
Present the instruction as chunks
Chunkcounter =chunkcounter+1
Until chunkcounter =7

Start the clock to monitor the time taken to complete the task

At any instant X of the given instruction

Instantaneous load = Load experienced at that instant

If skin conductivity is high then load =high
Set Instantaneous Load > 7
Else
If skin conductivity is high then load =low
Set Instantaneous Load <7

Else
If skin conductivity is moderate then load =high
Set Instantaneous Load = 4

Free Capacity = Peak Load - Instantaneous Load

For a given task

Accumulated load = Sum of Instantaneous load of various chunks of instruction/number of instructions

Average load = Accumulated Load/No of chunks

At any instant of performing task

If instantaneous load >peak load then Display the message" Cognitive Load has happened at that instant" Customize the Instruction by informing the tutorial module.

If average load >peak load then Display the message"Cognitive Load has happened for the task" Inform the tutorial model to tailor the remedial measure.

else

If instaneous load < peak load then Display the message" No Cognitive Load has happened at that instant" Continue to next Instruction by informing the tutorial module.

If average load < peak load then Display the message"No Cognitive Load has happened for the task" Inform the tutorial model to proceed to the new task.

5 Tutorial module

This module is aimed at tailoring the instruction. The tutorial module will use the fractal tree approach to present the instruction. Fractal tree is a good metaphor for presenting the instruction as the learner will be exploring only the branch which is of interest at the point of time.So it reduces the load avoiding excess information being presented to the learner.The following section explains on the program representation using a fractal tree.The choice of the metaphor fractal tree could be justified on the basis of it is based on exploratory learning.The information space is very large in learning a programming. The learner will be exploring the information space not as a whole but as a partial information space, thereby reducing the load. [15]

When talking about programming language the basic constructs that is used in a program may be defined as declarations which may be variable or constants or sometimes assignment statements. Another common construct that is used is looping statement such as if..then..else, switch, while etc. Another set of instructions included is the usage of library functions and constructs like classes, objects. We also use some constructs like methods which are similar to functions. The following example of a C++ program is considered to define the chunks

```
// This program adds two numbers
and prints their sum.
#include <iostream>

int main()
{
  int a = 123;
  int b (456);
  int sum;

  sum = a + b;

  std::cout << "The sum of " << a
<< " and " << b << " is " << sum
<< "\n";

  return 0; }
```

If we consider the above program we can present the program to the learner in various chunks namely library function, main function .The main function can be further divided as declarations namely int a,b,sum .It also can be divided as calculation part which is the step of adding the two numbers. The final part can be result section which is the display of the output result. So in generic we classify and program as shown below in Figure 9.

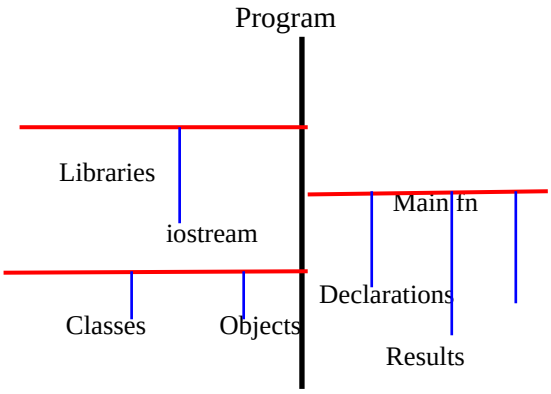

Figure 9: Presentation of program using fractal tree.

The above proposed mechanism of using fractal tree will help in schema acquisition or in short helps in meaningful organization of information as chunks. Each branch of the fractal tree is explored step by step in order to ensure that the information presented at any instant in the working memory is no more than seven. The idea of using the fractal tree can help in reducing the load since the information is presented in form of fractals which can be termed as chunks.

The performance model helps in bringing up with the analysis of the load based on the learning outcomes. The learning outcome can be related as the measure of difficulty faced by the student.We also can assume that measure of difficulty could indicate the cognitive overload happened if any.
This model will be designed to take the quiz or self assessment questions on each learning outcome.

6 Conclusion and future work

We observe that the currently used software visualization tools to learn programming has reduced the load while learning, but without taking into consideration the individual differences in learning styles as there is no component to measure the cognitive load. Both the physiological and non physiological measures are generally used to measure cognitive load. The proposed framework will help overcoming the major hurdle of learning. The innovative step in this framework is to develop a mechanism to measure the load. Physiological measures like Galvanic skin response and in addition to non physiological measure such as self rating by the learner will provide the approach to determine the load experienced by the learner. If any overload happens, leads to customization of the instruction that results in reduced load. The framework is currently implemented as prototype. The results of adopting the framework will be precise once the prototype is used in class room /self learning instruction. A tool is also being developed based on the proposed framework.

References

[1] Baddeley, A.D., & Wilson, B. A. (2002). Prose recall and amnesia: implications for the structure of working memory. Neuropsychologia, Vol 40, pp 1737-1743

[2] Ben-Basset Levy et. al (2003). The Jeliot 2000 program animation system, Computers & Education Vol 40, No 1, Pg 1-15

[3]Curts.S&Martha.E.Crosby (2005).Assessing Cognitive load with Physiological sensors. Proceedings of the 38th Hawaii International Conference of System Science-2005

[4].F.Paas et.al (2003).Cognitive Load measurement as a Means to Advance Cognitive Load Theory. Educational Psychologist, Vol 38, No 1, pp 63-71.

[5] Garner.S (2002).COLORS for Programming: A system to support the Learning of Programming. Proceedings of Informing science.pp 533-541

[6] G.Stuart (2002).Reducing the cognitive load for novice programmers. Proceedings of the ED-Media 2002 World Conference on Educational Multimedia Hypermedia and Telecommunications, Colorodo

[7] Lin T et. al (2003). Cognitive Trait Model- A supplement to Performance Based Student Models.Proceedings of the International Conference on Computers in Education.Dec 2-5,pp 629-632,VA,USA

[8]Marcus.N (1996).Understanding Instructions. Journal of Educational Psychology, Vol 88, pp 49-63

[9] Kashihara A et.al(2000).A Cognitive Load Reduction Approach to Exploratory and Its Application to an Interactive Simulation-based Learning System.Journal of Educational Multimedia and Hypermedia,9(3),pp 253-276

[10] M.Kolling et.al (2003). The BlueJ system and its pedagogy. Journal of Computer Science Education, Special Issue on Learning and Teaching Object Technology,Vol 13, No 4, Dec 2003

[11]Moreno,A,Myller.N.,Sutinen,E.,Ben-Ari, (2004) Visualizing Programs with Jeliot 3, Proceedings of Advanced Visual Interfaces,AVI 2004, pp 373-376

[12]R.A.Jeffreis et.al. (1981)The processes involved in designing software.In J.R. Anderson(Ed), Cognitive skills and their acquisition.Lawerence Erlbaum Associates, Hillsdale, NJ, pp 255-283.

[13]R.Guindon (1990) Design the design process: exploiting opportunistic thoughts. Human Computer Interaction.Vol 5, pp 305-344

[14] Sykes, E. R and Franek, F. (2003). "An Intelligent Tutoring System Prototype for Learning to Program Java", In *Proceedings of the 3rd IEEE International Conference on Advanced Learning Technologies* (ICALT '03), Athens, Greece, July, 2003, pp 485-486

[15] Teong.J et.al (2005).Visualizing Hierarchies &collection structures with Fractal trees. Proceeding of the Sixth Mexican International Conference on Computer Science (ENC'05)

[16]T.J.McGill &S.E.Volet (1997). A conceptual framework for analyzing student's knowledge of programming. *Journal of Research on Computing in Education*. Vol 29, No 3, pp 276-297

[17] Yu Shi et.al (2007).Galvanic Skin Response (GSR) as an index of Cognitive Load. Proceedings of CHI 2007, Sanfransisco,pp 2651-2656